



TUGAS AKHIR TF141581

**PERANCANGAN SISTEM PENGENDALIAN
LEVEL DENGAN MENGGUNAKAN *INTERNAL
MODEL CONTROL* (IMC) PI PADA *STEAM
DRUM* DI PTPN X KREMBUNG**

GEBY NUR HAMZAH
NRP 2414 106 026

Dosen Pembimbing :
Ir. Ya'umar, M.T.

Departemen Teknik Fisika
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember
Surabaya
2017



FINAL PROJECTTF141581

**PERANCANGAN SISTEM PENGENDALIAN
LEVEL DENGAN MENGGUNAKAN *INTERNAL
MODEL CONTROL (IMC)* PI PADA *STEAM
DRUM* DI PTPN X KREMBUNG**

**GEBY NUR HAMZAH
NRP 2414 106 026**

**Advisor Lecturer :
Ir. Ya'umar, M.T.**

**Department of Engineering Physics
Faculty of Industrial Technology
Sepuluh Nopember Institute of Technology
Surabaya
2017**

PERNYATAAN BEBAS PLAGIASI

Saya yang bertanda tangan di bawah ini :

Nama : Geby Nur Hamzah

NRP : 2414 106 026

Departemen : Teknik Fisika FTI-ITS

Dengan ini menyatakan bahwa tugas akhir saya yang berjudul :

“PERANCANGAN SISTEM PENGENDALIAN *LEVEL* DENGAN MENGGUNAKAN *INTERNAL MODEL CONTROL* (IMC) PI PADA *STEAM DRUM* DI PTPN X KREMBUNG” adalah bebas dari plagiasi. Apabila pernyataan ini terbukti tidak benar, maka saya bersedia menerima sanksi sesuai ketentuan yang berlaku.

Demikian surat pernyataan ini saya buat dengan sebenar-benarnya.

Surabaya, Juli 2017
Yang membuat pernyataan

Geby Nur Hamzah

(Halaman ini sengaja dikosongkan)

LEMBAR PENGESAHAN

**PERANCANGAN SISTEM PENGENDALIAN *LEVEL*
DENGAN MENGGUNAKAN *INTERNAL MODEL*
CONTROL (IMC) PI PADA *STEAM DRUM* DI PTPN X
KREMBUNG**

TUGAS AKHIR

Oleh:

Geby Nur Hamzah

NRP : 2414 106 026

Surabaya, Juli 2017

Mengetahui / Menyetujui

Pembimbing



Ir. Yakumar, M. T.

NIP. 19540406 198103 1 003

Kepala Departemen Teknik Fisika



Agus Muhammad Hatta, ST, MSi, Ph.D

NIP. 19780902 200312 1 002

(Halaman ini sengaja dikosongkan)

**PERANCANGAN SISTEM PENGENDALIAN *LEVEL*
DENGAN MENGGUNAKAN *INTERNAL MODEL*
CONTROL (IMC) PI PADA *STEAM DRUM* DI PTPN X
KREMBUNG**





TUGAS AKHIR

Diajukan Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Teknik
pada
Bidang Studi S-1 Teknik Fisika
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember

Oleh :

Geby Nur Hamzah
NRP. 2414 106 026

Disetujui oleh Tim Penguji Tugas Akhir :

1. Ir. Ya'umar, M.T. (Pembimbing)
2. Totok Ruki Biyanto, Ph.D. (Ketua Penguji)
3. Rahmadiansah, S.T., M.T. (Penguji 1)
4. Detak Yan Pratama, S.T., M.Sc. (Penguji 2)

SURABAYA
Juli, 2017

(Halaman ini sengaja dikosongkan)

PERANCANGAN SISTEM PENGENDALIAN LEVEL DENGAN MENGGUNAKAN *INTERNAL MODEL CONTROL* (IMC) PID PADA *STEAM DRUM* DI PTPN X KREMBUNG

Nama Mahasiswa : Geby Nur Hamzah
NRP : 2414 106 026
Program Studi : S-1 Teknik Fisika FTI-ITS
Dosen Pembimbing : Ir. Ya'umar, M.T.

Abstrak

Steam drum adalah suatu alat yang berfungsi untuk memisahkan antara air dan uap sebelum masuk pada *superheater*. *Steam drum* bekerja dengan memanfaatkan air yang masuk pada *steam drum* melalui *economizer*, kemudian di panaskan sehingga dapat memisahkan antara air dan uap. Pengendalian *level* pada *steam drum* perlu dilakukan untuk menjaga kestabilan pasokan air yang masuk ke *steam drum*. Perancangan sistem pengendalian dengan metode *Internal Model Control* (IMC) bertujuan agar respon pengendali memiliki performansi yang lebih baik daripada pengendali PID. Dari hasil simulasi didapatkan bahwa pengendali IMC memiliki respon yang lebih baik pada saat di uji pengendali *close loop*, dan mampu mencapai *set point* pada saat uji *tracking set point*. Parameter *tuning* IMC menunjukkan respon terbaik pada saat λ 0,1. Parameter respon pengendali IMC yang di dapatkan yaitu, *Maksimum overshoot* = 0,00625%, *Settling time* = 7,545023s, *Error steady state* = 0,000025 dan *Integral absolute error* = 8,764348.

Kata kunci:

Steam drum, internal model control, PID

(Halaman ini sengaja dikosongkan)

**DESIGNING LEVEL CONTROL SYSTEM USING
INTERNAL MODEL CONTROL (IMC) PID ON STEAM
DRUM IN PTPN X KREMBUNG**

Nama Mahasiswa : Geby Nur Hamzah
NRP : 2414 106 026
Program Studi : S-1 Teknik Fisika
Dosen Pembimbing : Ir. Ya'umar, M.T.

Abstrac

Steam drum is a tool that serves to separate between water and steam before entering the superheater. Steam drum works by utilizing water entering the steam drum through the economizer, then in the heat so as to separate between water and steam. Level control on the steam drum needs to be done to maintain the stability of water supply into steam drum. The design of control system with Internal Model Control method (IMC) aims to controller response has better performance than PID controller. From the simulation results obtained that the IMC controller has a better response at the time of the close loop control test, and able to reach the set point at the time of tracking point set test. IMC tuning parameters show best response at λ 0.1. IMC controller response parameters that get, that is, Maximum overshoot = 0,00625%, Settling time = 7,545023s, Error steady state = 0,000025 and Integral absolute error = 8,764348

Keyword :

Steam drum, internal model control, PID

(Halaman ini sengaja dikosongkan)

KATA PENGANTAR

Puji syukur penulis panjatkan kepada Allah S.W.T, karena rahmat dan hikmat-Nya sehingga penulis diberikan kesehatan, kemudahan, dan kelancaran dalam menyusun laporan tugas akhir ini. Tidak lupa juga penulis menyampaikan ucapan terima kasih kepada keluarga dan para sahabat. Oleh karena dukungan mereka, penulis mampu menyusun laporan tugas akhir yang berjudul:

“PERANCANGAN SISTEM PENGENDALIAN *LEVEL* DENGAN MENGGUNAKAN *INTERNAL MODEL CONTROL* (IMC) PI PADA *STEAMDRUM* DI PTPN X KREMBUNG”

Tugas akhir ini merupakan salah satu persyaratan akademik yang harus dipenuhi dalam Program Studi S-1 Teknik Fisika FTI-ITS. Penulis menyampaikan terima kasih yang sebesar-besarnya kepada:

1. Bapak Agus Muhamad Hatta, ST, MSi, Ph.D. selaku kepaladepartemen Teknik Fisika ITS.
2. Segenap keluarga yang telah memberikan dukungan penuh dalam penyelesaian tugas akhir ini.
3. Bapak Ir. Ir. Ya'umar, M.T. selaku dosen pembimbing tugas akhir ini, yang selalu memberikan semangat dan ide-ide baru.
4. Bapak Ir. Purwadi Agus Darwito, M.Sc. selaku dosen wali selama 2.5 tahun masa perkuliahan.
5. Bapak Totok Ruki Biyanto Ph.D. selaku ketua penguji dan kepala Laboratorium Rekayasa Instrumentasi
6. Bapak Rahmadiansah, S.T., M.T. dan Bapak Detak Yan Pratama, S.T., M.Sc. selaku dosen penguji.
7. Kepada PTPN X Krembung tempat dimana penulis mengambil data Tugas Akhir.
8. Rekan-rekan LJ TF 2015 selaku rekan seperjuangan dari awal kuliah hingga tersusunnya laporan tugas akhir ini.
9. Rekan-rekan Workshop Instrumentasi yang selalu menyediakan tempat untuk mengerjakan tugas akhir ini.

10. Segenap Bapak/Ibu dosen pengajar di departemen Teknik Fisika - ITS.

Penulis menyadari bahwa mungkin masih ada kekurangan dalam laporan ini, sehingga kritik dan saran penulis terima. Semoga laporan ini dapat berguna dan bermanfaat bagi penulis dan pihak yang membacanya.

Surabaya, 6 Juni 2017

Penulis

DAFTAR ISI

HALAMAN JUDUL	i
LEMBAR PENGESAHAN	v
LEMBAR PENGESAHAN	vii
ABSTRAK	xi
ABSTRACT	xiii
KATA PENGANTAR	xv
DAFTAR ISI	xvii
DAFTAR GAMBAR	xxi
DAFTAR TABEL	xxiii

BAB I PENDAHULUAN

1.1 Latar Belakang	1
1.2 Perumusan Masalah	2
1.3 Tujuan Tugas Akhir	2
1.4 Batasan Masalah	3

BAB II TEORI PENUNJANG

2.1 <i>Steam Drum</i> pada <i>Bolier</i> Cheng-Chen PTPN X Krembung ..	5
2.2 Pengendali PID	7
2.2.1 Pengendali Proporsional (P)	7
2.2.2 Pengendali Integral (I)	8
2.2.3 Pengendali Diferensial (D)	10
2.3 <i>Internal Model Control</i> (IMC)	12
2.3.1 Model Based IMC	13
2.3.2 Struktur IMC-PID	18
2.4 <i>Respon Transient</i>	23

BAB III METODOLOGI PENELITIAN

3.1 Alur Penelitian	25
3.2 Studi Literatur	26
3.3 Pengambilan Data	26
3.4 Model Matematis <i>Steam Drum</i>	27
3.5 Uji <i>Open Loop Gain Plant</i>	36
3.5.1 Hasil Simulasi Uji <i>Open Loop Plant</i>	36

3.6	Uji <i>Close Loop Plant</i>	37
3.6.1	Hasil Simulasi Uji <i>Close Loop Plant</i>	38
3.7	Uji <i>Disturbance</i>	39
3.8	Pengendali IMC	40
3.8.1	Uji Pengendali Tanpa <i>Disturbance</i>	40
3.8.2	Uji Pengendali Dengan <i>Disturbance</i>	43
3.9	Perbandingan Respon Sistem.....	44

BAB IV ANALISA DATA DAN PEMBAHASAN

4.1	Hasil Perancangan Pengendali IMC	47
4.1.1	Hasil Perancangan Pengendali Tanpa <i>Disturbance</i>	47
4.1.2	Hasil Perancangan Pengendali Dengan <i>Disturbance</i>	48
4.2	Hasil Perancangan IMC Dengan <i>Tuning</i> Acak.....	49
4.3	Hasil Perbandingan Respon Pengendali	51
4.4	Pengujian <i>Tracking Set Point</i> Pengendali IMC	52
4.3	Pembahasan	54

BAB V PENUTUP

5.1	Kesimpulan	57
5.2	Saran	57

DAFTAR PUSTAKA

LAMPIRAN

DAFTAR GAMBAR

Gambar 2.1	<i>P&ID Steam drum pada Boiler Cheng-Chen</i>	6
Gambar 2.2	Diagram Blok Pengendali P	7
Gambar 2.3	Diagram Blok Pengendali	9
Gambar 2.4	Diagram Blok Pengendali D	10
Gambar 2.5	Diagram Blok Pengendali PID	11
Gambar 2.6	Proses <i>Open Loop</i> Pengendali IMC	13
Gambar 2.7	Struktur <i>Internal Model Control</i>	16
Gambar 2.8	Struktur IMC-PID	17
Gambar 2.9	<i>Feedback-feedforward</i> dengan IMC-PID	18
Gambar 2.10	Grafik Respon <i>Transient</i>	24
Gambar 2.11	Penentuan Nilai <i>Integral Absolute Error</i>	24
Gambar 3.1	<i>Flowchart</i> Pengerjaan Tugas Akhir	25
Gambar 3.2	<i>Process flow diagram steam drum</i>	26
Gambar 3.3	<i>P&ID steam drum</i>	27
Gambar 3.4	Luas <i>Steam Drum</i>	29
Gambar 3.5	Luas <i>Steam Drum</i> Sampling	30
Gambar 3.6	Uji <i>Open Loop Gain Plant</i>	36
Gambar 3.7	Grafik Uji <i>Open Loop Gain Plant</i>	37
Gambar 3.8	Uji <i>Close Loop Plant</i>	37
Gambar 3.9	Grafik Uji <i>Close Loop Plant</i>	38
Gambar 3.10	Uji <i>Disturbance</i>	39
Gambar 3.11	Simulasi <i>Disturbance</i>	39
Gambar 3.12	Hasil Simulasi Uji <i>Disturbance</i>	40
Gambar 3.13	Uji Pengendali Tanpa <i>Disturbance</i>	43
Gambar 3.14	Simulasi Uji Pengendali Dengan <i>Disturbance</i>	44
Gambar 3.15	Simulasi <i>Subsystem Disturbance</i>	44
Gambar 3.16	Simulasi Perbandingan Pengendali IMC dan PI	45
Gambar 4.1	Respon Pengendali IMC Tanpa <i>Disturbance</i>	47
Gambar 4.2	Respon Pengendali IMC Dengan <i>Disturbance</i>	48
Gambar 4.3	Respon Pengendali IMC Dengan <i>Disturbance</i>	49

Gambar 4.4	Perancangan Pengendali IMC <i>Tuning</i> $\lambda = 0,01$ dan $\lambda = 0,05$	50
Gambar 4.5	Hasil Perancangan pengendali IMC <i>Tuning</i> λ $= 0,01$ dan $\lambda = 0,05$	50
Gambar 4.6	Respon Perbandingan Pengendali IMC dan PID.....	51
Gambar 4.7	Simulasi Uji <i>Tracking Set Point</i> Pengendali IMC Dengan Variasi 4 <i>Lambda</i>	53
Gambar 4.8	Hasil <i>Respon</i> Uji <i>Tracking Set Point</i> Pengendali IMC Dengan Variasi 4 <i>Lambda</i>	54

DAFTAR TABEL

Tabel 3.1	<i>Data Input dan Output Steam Drum</i>	27
Tabel 3.2	<i>Data Spesifikasi Dimensi Steam Drum</i>	27
Tabel 4.1	<i>Parameter Pengendali Tanpa Disturbance</i>	48
Tabel 4.2	<i>Parameter Pengendali Dengan Disturbance</i>	49
Tabel 4.3	<i>Parameter Tuning IMC $\lambda = 0,01$ dan $\lambda = 0,05$</i>	51
Tabel 4.4	<i>Perbandingan Parameter Respon Pengendali</i>	52
Tabel 4.5	<i>Perbandingan Parameter Tuning Pengendali</i>	52

(Halaman ini sengaja dikosongkan)

BAB I

PENDAHULUAN

1.1 Latar Belakang

Pada pengendalian *level steam drum* perlu di kendalikan, untuk menjaga ketinggian *level* air berada pada ketinggian setengah dari tinggi *steam drum*. Hal ini penting di lakukan poengendalian karena, apabila level air terlalu rendah dari tinggi steam drum maka, boiler akan mengalami *overheating*. Apabila *level* air pada *steam drum* terlalu tinggi, maka gelembung-gelembung air yang berada pada *steam drum* terbawa ke turbin, sehingga akan mengurangi *lifetime* pada turbin. *Steam drum* pada boiler jenis Cheng-Chen di PTPN X Krembung, menggunakan kendali PIDcontrol system. Kendala yang sering dialami adalah ketidakstabilan respon pengendalian, dimana pengendali PID yang di lakukan tidak memenuhi *set pont*. Pada *variable* proses yang di kendalikan yaitu *temperature, pressure, level, flow*. *Variable-variable* pada *steam drum* perlu di kendalikan agar *set point* yang nantinya masuk pada turbin terpenuhi.

Sistem umpan balik dengan pengendali PID, yang sering diterapkan pada sistem *control* di industri saat ini, mempunyai beberapa kelemahan. Diantaranya yaitu, apabila ada gangguan yang tidak diduga pada *system* pengendalian. Gangguan yang terjadi pada pengendalian akan mempengaruhi *output*. Sehingga *output* menjadi tidak sesuai dengan *set point*, dan mempengaruhi waktu dalam mencapai *set point*. IMC adalah salah satu metode *control*, yang menggunakan model dari *plant* yang ingin dikontrol. IMC dirancang pada suatu pengontrol umpan balik, untuk membuat keluaran suatu proses yang stabil, dan memberikan respon sesuai yang diinginkan terhadap perubahan *set point*, dan juga mengatasi pengaruh gangguan yang langsung masuk pada keluaran proses.

Teknik pengendalian IMC-PID yang sering diterapkan pada *power plant*, yakni dengan melakukan *tuning* terhadap perubahan *set point*. Pada kenyataannya, tidak hanya *tuning* berdasarkan perubahan *set point* saja, tetapi juga dapat menghasilkan

respond yang tepat. Tetapi *tuning* berdasarkan nilai *disturbance* juga tepat untuk digunakan. Pada *tuning disturbance* tersebut dapat mengurangi nilai gangguan yang langsung masuk pada keluaran sebuah proses (Biyanto, 2013). Pada penelitian sebelumnya yang dilakukan oleh Moamar Arief, dengan judul “*Perancangan Sistem Pengendalian Level Menggunakan Metode Internal Model Control Berbasis PID Pada Deaerator*” telah dilakukan pengendalian *power plant*. Yang bertujuan untuk menghasilkan *respond* yang sesuai dengan *set point*, dan mengurangi tingkat gangguan yang masuk pada proses. Namun hal ini hanya diterapkan pada unit *plant daerator*. Oleh karena itu, dalam tugas akhir ini akan dirancang sebuah pengendalian *level* dengan menggunakan *Internal Model Control* (IMC) PID Pada *Steam Drum* di PTPN X krembung.

1.2 Perumusan masalah

Dari uraian latar belakang tersebut, didapatkan beberapa permasalahan, yaitu:

- a. Bagaimana merancang *system* pengendali *Internal Model Control* (IMC).
- b. Bagaimana performansi metode IMC maupun PID *control system* pada *Steam Drum*.

1.3 Tujuan Tugas Akhir

Tujuan dari penelitian ini adalah menjawab permasalahan yang telah dikemukakan, yakni:

- a. Melakukan perancangan *system* pengendali *Internal Model Control* (IMC).
- b. Menganalisa performansi yang dihasilkan dari pengendalian *level* pada *Steamdrum*.

1.4 Batasan Masalah

Batasan masalah dalam tugas akhir ini antara lain:

1. Proses yang di kendalikan ialah *level steam drum* pada *boiler* Cheng-Chen di PTPN X Krembung.
2. Simulasi menggunakan MATLAB R2015b.
3. Pemodelan *plant* di lakukan dengan menggunakan *white box model level steam drum*.

Halaman ini sengaja dikosongkan

BAB II

TEORI PENUNJANG

Pada bab II ini berisi teori penunjang tentang segala hal yang berkaitan dengan judul tugas akhir, sehingga nantinya dapat membantu pola berpikir dalam menyelesaikan permasalahan yang terdapat di dalam penulisan. Teori penunjang tersebut didapatkan melalui *referensi text book, manual book, jurnal ilmiah* yang telah dipublikasikan

2.1 Steam Drum pada Boiler Drum Jenis Cheng-Chen di PTPN X Krembung

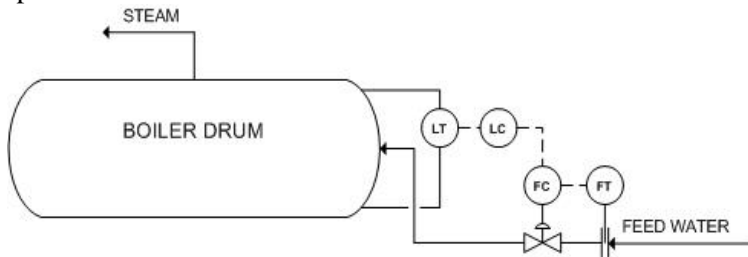
Steam drum merupakan salah satu komponen pada *boiler* pipa air, yang berfungsi sebagai *reservoir* campuran air dan uap air, dan juga berfungsi untuk memisahkan uap air dengan air, pada proses pembentukan uap *superheater*. *Steam Drum* pada *boiler* PTPN X Krembung berfungsi untuk:

1. Mengatur tinggi permukaan air untuk mencegah terjadi kekurangan air, pada saat *boiler* beroperasi yang dapat menyebabkan *overheating* pada pipa *boiler*.
2. Menampung air, yang nantinya akan dipanaskan pada pipa-pipa penguap (*wall tube*), dan menampung uap air dari pipa-pipa penguap, sebelum dialirkan menuju *superheater*.
3. Memisahkan uap dan air yang telah dipisahkan di ruang bakar (*furnace*).
4. Mengatur kualitas air *boiler*, dengan membuang kotoran-kotoran terlarut di dalam *boiler* melalui *continious blowdown*.

Sedangkan *Superheater* merupakan suatu komponen dalam *boiler*, yang digunakan untuk memanaskan lanjutan uap saturasi, sampai menghasilkan uap yang benar-benar kering^[8]. Pada pengendalian temperatur ini, temperatur harus dijaga supaya selalu mencapai *set point*, agar sesuai dengan temperatur yang digunakan untuk memutar turbin. *Steam* yang didapat dari *boiler* masih berupa uap basah, uap basah tersebut kemudian dialirkan menuju *Superheater* untuk dipanaskan kembali. Tapi sebelum masuk *superheater*, *temperature* disemprotkan air sampai suhu

optimal pada saat kerja normal, atau saat terjadi kenaikan beban. Kemudian melalui *superheater* akan dihasilkan uap kering, agar bisa menjalankan *turbin* dengan baik. Bila temperatur melebihi *set point*, maka akan menjadi uap kering yang membahayakan bagi turbin, serta bahan bakar akan terbuang sia-sia. Oleh karena itu, *desuperheater* (*superheater spray*) akan mengeluarkan air yang berfungsi untuk menurunkan suhu *hot steam* hingga mencapai *set point*.

Pada pembentukan uap, dari uap basah sampai uap kering, hal ini harus memperhatikan kinerja dari *steam drum*. Dimana *steam drum* memiliki peranan yang sangat penting, untuk tercapainya uap yang di inginkan *turbin*. *Level* air pada *steam drum* harus di kendalikan, agar selalu memenuhi *set point*. Di mana sistem kerja steam drum pada boiler di PTPN X Krembung, seperti Gambar 2.1.



Gambar 2.1 P&ID *Steam drum* pada Boiler Cheng-Chen

Liquid yang masuk pada *boiler drum* dari *feed water*, sebelumnya telah di panaskan terlebih dahulu pada *economizer*. Kemudian masuk pada *drum* dalam keadaan sudah panas, *Liquid* masuk dalam drum yang sudah panas akan di panaskan kembali, sehingga terbentuk uap. Namun uap pada kluaran drum masih bersifat basah, dan akan di panaskan kembali melalui *superheater*.

Dimana;

LT = *Level transmitter*

LC = *Level control*

FC = *Flow control valve*

FT = *Flow transmitter*.

2.2 Pengendali PID

Sesuai dengan namanya, pengendali ini terdiri atas pengendali *Proportional (P)*, *pengendali Integral (I)* dan *pengendalian Diffrential (D)*. Masing-masing pengendali mempunyai karakter yang unik.

2.2.1 Pengendali Proporsional (P)

Pengendali *proporsional* memiliki 2 parameter, *pita proporsional (proportional band)* dan konstanta *proporsional*. Daerah kerja pengendali efektif dicerminkan oleh *pita proporsional*^[2], sedangkan konstanta *proporsional* menunjukkan nilai faktor penguatan terhadap sinyal kesalahan, K_p . Hubungan antara *pita proporsional (PB)* dengan konstanta *proporsional* (K_p) ditunjukkan secara presentasi oleh Persamaan ^[2].

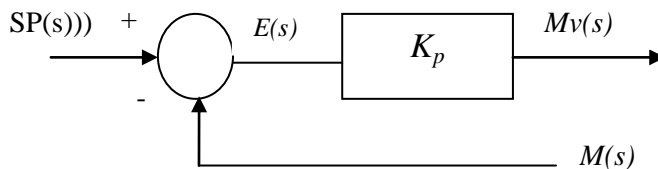
$$PB = \frac{1}{K_p} \times 100\% \quad (2.1)$$

Dengan:

PB = *Proportional band*

K_p = *Gain proces*

Diagram blok pengendali *proportional* ditunjukkan seperti pada Gambar 2.2.



Gambar 2.2Diagram Blok Pengendali P^[2]

Dengan:

$Sp(s)$ = *Set point*

$E(s)$ = *Error*

K_p = *Gain proses*

$Mv(s)$ = *Manipulated variable*

$M(s)$ = *Manipulated feedback*

Pengguna pengendali *proporsional* memperhatikan hal-hal berikut:

1. Kalau nilai K_p kecil, pengendali *proporsional* hanya mampu melakukan koreksi kesalahan yang kecil, sehingga akan menghasilkan *respon system* yang lambat.
2. Kalau nilai K_p dinaikkan, *respon sistem* menunjukkan semakin cepat mencapai keadaan mantapnya.
3. Namun jika nilai K_p diperbesar sehingga mencapai harga yang berlebihan, akan mengakibatkan *system* bekerja tidak stabil, atau *respon sistem* akan berosilasi

2.2.2 Pengendali Integral (I)

Pengendali *integral* memiliki karakteristik seperti halnya sebuah integral. Keluaran pengendali sangat dipengaruhi oleh perubahan yang sebanding dengan nilai sinyal kesalahan. Keluaran pengendali ini merupakan jumlahan yang terus menerus dari perubahan masukannya. Kalau sinyal kesalahan tidak mengalami perubahan, keluaran akan menjaga keadaan seperti sebelum terjadinya perubahan masukan. *Transfer function* dari unit pengendali integral adalah sebagai berikut [Gunterus, Frans, 1994]:

$$u = \frac{1}{T_I} K_c \int e \cdot dt \quad (2.2)$$

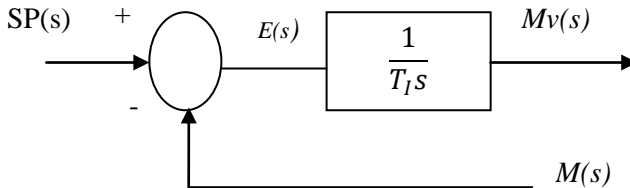
dimana,

T_I = *Integral time*

e = *Error* (input dari unit control)

K_c = *Gain* dari pengendali

Diagram blok pengendali *integral* ditunjukkan oleh Gambar 2.3 dibawah ini:



Gambar 2.3 Diagram Blok Pengendali^[4]

Dengan:

$Sp(s)$ = *Set point*

$E(s)$ = *Error*

T_i = *Integral time*

$Mv(s)$ = *Manipulated variable*

$M(s)$ = *Manipulated feedback*

Ketika digunakan, pengendali integral mempunyai beberapa karakteristik berikut ini:

1. Keluaran pengendali membutuhkan selang waktu tertentu, sehingga pengendali *integral* cenderung memperlambat respon.
2. Ketika sinyal kesalahan berharga nol, keluaran pengendali akan bertahan pada nilai sebelumnya.
3. Jika sinyal kesalahan tidak berharga nol, keluaran akan menunjukkan kenaikan atau penurunan yang dipengaruhi oleh besarnya sinyal kesalahan dan nilai K_i .
4. Konstanta *integral* K_i yang berharga besar akan mempercepat hilangnya *offset*. Tetapi semakin besar nilai konstanta K_i akan mengakibatkan peningkatan osilasi dari sinyal keluaran pengendali [Gunterus, 1994].

2.2.3 Pengendali Diferensial (D)

Keluaran pengendali *diferensial* memiliki sifat seperti halnya suatu operasi *derivatif*. Perubahan yang mendadak pada masukan kontroler, akan mengakibatkan perubahan yang sangat besar dan cepat. *Transfer function* dari pengendali *derivative* adalah sebagai berikut [Gunterus, 1994].

$$u = K_c T_D \frac{de}{dt} \quad (2.3)$$

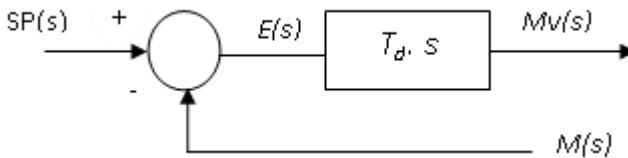
dimana,

K_C = Gain pengendali

e = Error

T_D = Derivative time

Diagram blok pengendali *diferensial* ditunjukkan oleh Gambar 2.4.



Gambar 2.4 Diagram Blok Pengendali D^[4]

Dengan:

$Sp(s)$ = Set point

$E(s)$ = Error

T_d = Derivative time

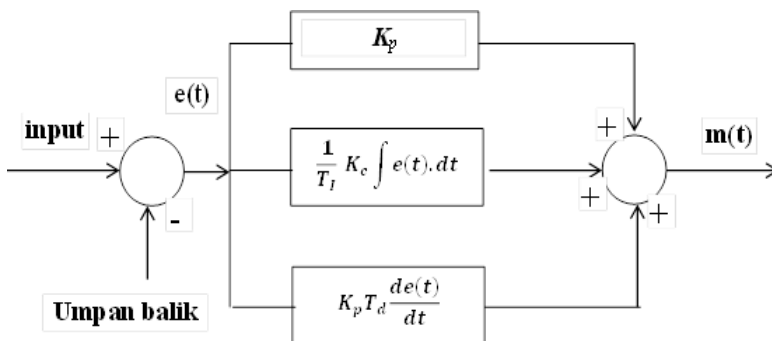
$Mv(s)$ = Manipulated variable

$M(s)$ = Manipulated feedback

Adapun untuk karakteristik pengendali *diferensial* adalah sebagai berikut:

1. Pengendali ini tidak dapat menghasilkan keluaran bila tidak ada perubahan pada masukannya (berupa sinyal kesalahan).
2. Jika sinyal kesalahan berubah terhadap waktu, maka keluaran yang dihasilkan pengendali tergantung pada nilai T_d dan laju perubahan sinyal kesalahan.
3. Pengendali *diferensial* mempunyai suatu karakter untuk mendahului, sehingga pengendali ini dapat menghasilkan koreksi yang signifikan sebelum pembangkit kesalahan menjadi sangat besar. Jadi pengendali *diferensial* dapat mengantisipasi pembangkit kesalahan, memberikan aksi yang bersifat korektif, dan cenderung meningkatkan stabilitas sistem.

Untuk menutupi kekurangan dari masing-masing pengendali, ketiga pengendali tersebut dapat saling dikombinasikan menjadi pengendali PI, PD atau PID. Dalam kombinasi PID, diagram bloknya menjadi seperti pada Gambar 2.5.



Gambar 2.5 Diagram Blok Pengendali PID^[4]

Keluaran pengendali PID merupakan jumlahan dari keluaran pengendali proporsional, keluaran pengendali *integral* dan keluaran pengendali *diferensial*. Gambar 2.5 menunjukkan hubungan input dan output pada pengendalian PID [Gunterus, Frans, 1994].

$$u(t) = K_c(e(t) + \frac{t}{T_I} \int_0^t e(t)dt + T_D \frac{de}{dt}) \quad (2.4)$$

Dimana,

$u(t)$ = Sinyal control

$e(t)$ = Error

K_c = Gain controller

T_I = Integral time

T_D = Derivative time

Karakteristik pengendali PID sangat dipengaruhi oleh kontribusi besar dari ketiga parameter yaitu, P, I dan D. Penyetelan konstanta K_p , T_i , dan T_d akan mengakibatkan penonjolan sifat dari masing-masing elemen. Satu atau dua dari ketiga konstanta tersebut dapat *disetting* lebih menonjol dibanding yang lain. Konstanta yang menonjol itulah akan memberikan kontribusi pengaruh pada *respon sistem* secara keseluruhan.

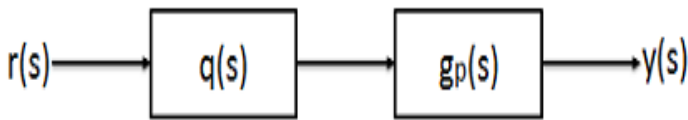
2.3 Internal Model Control (IMC)

Internal model control (IMC) adalah pengendalian yang hanya dapat dicapai, jika sistem memuat parameter dari proses yang akan dikendalikan, baik secara *implisit* maupun *eksplisit*^[6]. Dapat dikatakan juga bahwa, IMC merupakan suatu metode yang dirancang untuk mengendalikan umpan balik, dari keluaran suatu proses. Skema pengendali IMC menggunakan *invers* dari fungsi transfer *plant*, untuk mencari fungsi transfer dari pengendali. Jika skema pengendali yang digunakan berdasarkan pada model dari *plant*, maka memungkinkan untuk mendapatkan pengendalian yang sempurna. IMC merupakan pengembangan dari algoritma control, yang memanfaatkan *inverse* dari model proses, dengan mempunyai satu parameter *tuning* yaitu *filter factor* (λ) (Morari, 1986). Pada pengendalian *internal model control* ini dapat digunakan untuk mengendalikan nilai *disturbance*, yang langsung masuk kedalam proses. Metode *tuning* ini bisa digunakan

untuk mengendalikan nilai *set point*, pada *plant* yang akan dikendalikan.

2.3.1 Model Based IMC

Kendali IMC merupakan pengendali yang menggunakan model dari proses sebagai acuan dari pengendali IMC itu sendiri. Pada Gambar 2.6 di bawah ini merupakan gambar *system open loop control IMC*^[7]



Gambar 2.6 Proses *Open Loop* Pengendali IMC^[5]

$q(s)$ merupakan pengendali dari proses, g_p merupakan fungsi *transfer* dari *model* proses. Dari Gambar 2.6 didapatkan persamaan,

$$y(s) = g_p(s) q(s) r(s) \quad (2.5)$$

Dimana;

$r(s)$ = *Input*

$q(s)$ = Pengendali (*internal model control*)

$g_p(s)$ = *Gain proses*

$y(s)$ = *Output*

Untuk *first order process*, $g_p(s) = \frac{k_p}{\tau s + 1}$, maka untuk persamaan diatas menjadi,

$$y(s) = \frac{k_p k_q}{\tau s + 1} r(s) \quad (2.6)$$

Dimana;

$r(s)$ = *Input*

K_p = *Gain plant*

K_q = *Gain proses controller*

$y(s)$ = Output
 τs = Time constant

Persamaan diatas adalah persamaan apabila proses dalam keadaan *steady state* atau statis. Pada proses yang dinamik, pengendali IMC yang digunakan adalah,

$$q(s) = \frac{1}{g_p(s)} \quad (2.7)$$

Dimana;

$q(s)$ = Pengendali(*internal model control*)
 $g_p(s)$ = Gain proses

Untuk *first order process*, maka persamaan dari pengendali menjadi,

$$q(s) = \frac{\tau s + 1}{k_p} \quad (2.8)$$

Dimana;

$q(s)$ = Pengendali(*internal model control*)
 $k_p(s)$ = Gain plant
 τs = Time constant

Pada pengendali IMC, pengendali yang digunakan harus terdapat *denominator polynomial*^[7]. *Denominator polynomial* pada IMC adalah berupa *filter* dengan persamaan,

$$f(s) = \frac{1}{\lambda(s) + 1} \quad (2.9)$$

Dimana;

$f(s)$ = Filter
 λ = Parameter *tuning filter* (konstanta)

Filter ini berguna untuk *tuning* parameter pada pengendali IMC. Dimana λ adalah parameter *tuning* filter untuk pengendali IMC, nilai *lambda* berfungsi untuk mencari respon pengendali IMC yang paling cepat. Untuk mencari respon yang paling cepat, nilai λ harus diperkecil^[7]. Atau dengan kata lain, mencari nilai λ yang paling kecil untuk mendapatkan respon pengendali IMC yang paling cepat. Dari penambahan filter, didapatkan persamaan pengendali,

$$q(s) = \frac{1}{g_p(s)} = \frac{f(s)}{g_p(s)} \quad (2.10)$$

$$q(s) = \frac{\tau s + 1}{k_p} \cdot \frac{1}{\lambda(s) + 1} = \frac{\tau s + 1}{k_p(\lambda s + 1)} \quad (2.11)$$

Dimana:

$q(s)$ = Pengendali IMC

τs = Time constant plant

k_p = Gain plant

λ = Parameter *tuning* filter (konstanta)

Dengan λ yang optimal, maka akan di hasilkan performansi terbaik dari respon proses. Menurut rivera dalam seborg (2004 : 307) besar nilai λ yang optimal dapat diperoleh pada persamaan,

$$\lambda > 0,8\theta \quad (2.12)$$

$$\lambda < 0,1\tau \quad (2.13)$$

Dimana:

θ = Time delay

τ = Time constant

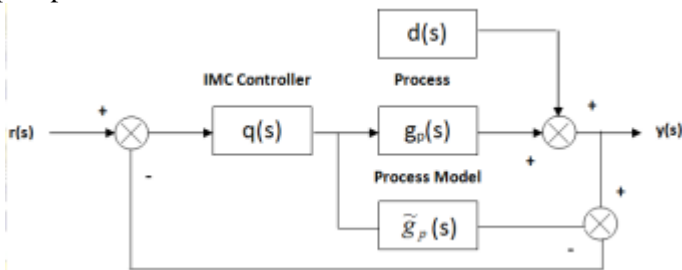
Di dalam sumber lain yang ditulis oleh Wyne Bequette^[7], untuk mendapatkan performansi yang terbaik maka nilai *lambda* filter harus sesuai dengan persamaan 2.14 dibawah ini.

$$\lambda > 1,5\theta \quad (2.14)$$

Sedangkan pada jurnal *simple analytic rules for model reduction and PID pengendali tuning* menurut Sigurd Skogestad^[6], besar nilai λ adalah seperti pada persamaan di bawah ini.

$$\lambda > 1 \quad (2.15)$$

Struktur pengendalian IMC dapat dirumuskan dalam struktur pengendali umpan balik standar. Struktur pengendali umpan balik standar ini, akan menghasilkan pengendali PID. sehingga dapat menggunakan parameter dari PID pengendali untuk menerapkan *advanced control* ini.^[6]



Gambar 2.7 Struktur *Internal Model Control*

Dimana;

$d(s)$ = Disturbance

$g_p(s)$ = Gain process

$q(s)$ = Internal model control

$r(s)$ = Set point

$\tilde{d}(s)$ = Estimated disturbance

$\tilde{g}_p(s)$ = Gain process model

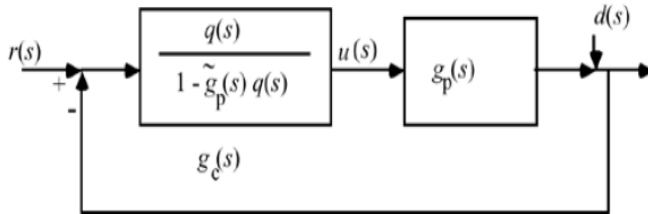
$u(s)$ = Manipulated input (controller output)

$\tilde{r}(s)$ = *Modified setpoint*

$y(s)$ = *Process output*

Fungsi *transfer model* yang di gunakan adalah sama dengan fungsi transfer *plant*. Dari gambar 2.7 struktur IMC dapat di lihat bahwa, fungsi *transfer model* yang di gunakan berfungsi untuk merubah sinyal *feedback* pada pengendali, menjadi sinyal *disturbance* yang sebelumnya merupakan sinyal proses, yang di tambah dengan sinyal *disturbance*. Oleh sebab itu, IMC sangat baik di gunakan untuk menghilangkan *disturbance* yang terdapat pada proses.

Untuk IMC-PID memiliki struktur *control* yang berbeda dengan IMC pada umumnya. Untuk lebih jelasnya dapat dilihat pada Gambar 2.8 berikut ini.



Gambar 2.8 Struktur IMC-PID^[6]

Dimana;

$d(s)$ = *Disturbance*

$g_p(s)$ = *Gain process*

$q(s)$ = *Internal Model control*

$r(s)$ = *Set point*

$\tilde{g}_p(s)$ = *Gain process model*

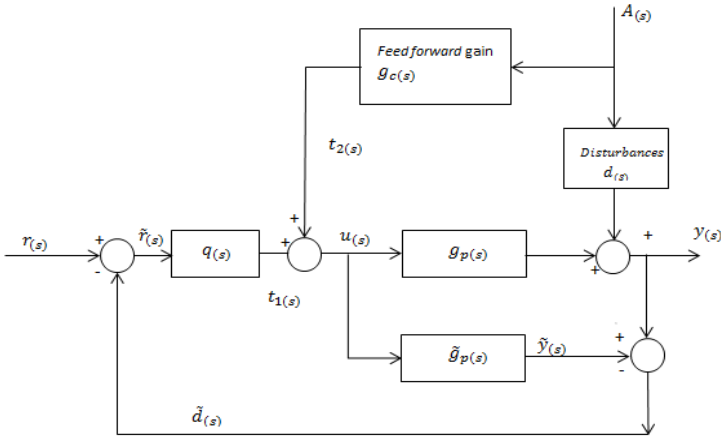
$u(s)$ = *Manipulated input (controller output)*

$\tilde{r}(s)$ = *Modified setpoint*

$y(s)$ = *process output*

2.3.2 Struktur IMC-PID

Pada struktur IMC-PID, invers dari *model plant* telah digunakan untuk menyusun pengendali. Sehingga pada simulasi tidak perlu memasukkan *invers model plant*. Berdasarkan sistem yang telah dirancang, dengan menggunakan strategi pengendali *feedback-feedforward* dengan *Internal Model Control* (IMC) berbasis PID, maka akan didapatkan diagram blok pengendali seperti Gambar 2.9 berikut;



Gambar 2.9 *feedback-feedforward* dengan IMC-PID ^[6]

Dari diagram blok diatas maka akan dapat dipresentasikan dengan penyederhanaan *loop control* seperti berikut ini.

$$\tilde{r}(s) = r(s) - \tilde{d}(s) \quad (2.16)$$

$$u(s) = t_1(s) + t_2(s) \quad (2.17)$$

$$y(s) = u(s) \cdot g_p(s) + d(s) \quad (2.18)$$

$$\tilde{d}(s) = y(s) - u(s) \cdot \tilde{g}_p(s) \quad (2.19)$$

$$t_2(s) = g_c(s) \cdot d(s) \quad (2.20)$$

$$t_1(s) = q(s) \cdot \tilde{r}(s) \quad (2.21)$$

$$\tilde{d}(s) = (g_p(s) - \tilde{g}_p(s)) u(s) + d(s) \quad (2.22)$$

$$\tilde{r}_{(s)} = r_{(s)} - \tilde{d}_{(s)} \quad (2.23)$$

Dimana;

$d(s)$ = *Disturbance*

$\tilde{d}(s)$ = *Estimated disturbance*

$g_p(s)$ = *Gain process*

$q(s)$ = *Internal model control*

$r(s)$ = *Set point*

$\tilde{g}_p(s)$ = *Gain processmodel*

$u(s)$ = *Manipulated input (controller output)*

$\tilde{r}(s)$ = *Modified setpoint*

$y(s)$ = *Process output*

$t1(s)$ = *Waktu 1*

$t2(s)$ = *Waktu 2*

Jika $g_p(s) = \tilde{g}_p(s)$ dan terdapat sinyal *disturbances* yang dapat mengganggu *plant proses*, maka akan didapatkan persamaan 2.24.

$$d_{(s)} = \tilde{d}_{(s)} \quad (2.24)$$

$$\tilde{g}_p(s) = \tilde{g}_{p+(s)} \cdot \tilde{g}_{p-(s)} \quad (2.25)$$

Dimana;

$d(s)$ = *Disturbance*

$\tilde{d}(s)$ = *Estimated disturbance*

$\tilde{g}_p(s)$ = *Gain processmodel*

Persamaan ini hanya digunakan untuk mendesain pengendali, proses *model* yang disimulasikan paralel dengan proses tetap menggunakan $\tilde{g}_{p(s)}$.

$$\tilde{q}_{(s)} = \frac{1}{\tilde{g}_{p-(s)}} \quad (2.26)$$

Dimana;

$\tilde{q}_{(s)}$ = *Estimate internal model control*

$\tilde{g}_p(s)$ = *Gain Process model*

Ditambahkan filter agar membuat pengendalian menjadi lebih baik, filter akan menentukan *respon* dari sistem. Filter akan membuat system menjadi lebih robust.

$$q_{(s)} = \tilde{q}_{(s)} \cdot f_{(s)} = \tilde{g}_{p-(s)}^{-1} f_{(s)} \quad (2.27)$$

$$f_{(s)} = \frac{1}{(\lambda_s + 1)^n} \quad (2.28)$$

Dimana;

$q(s)$ = *Internal model control*

$\tilde{q}_{(s)}$ = *Estimate internal model control*

$\tilde{g}_p(s)$ = *Gain process model*

$f_{(s)}$ = *Filter*

$\lambda_{(s)}$ = *Parameter tuning filter (konstanta)*

Jika membutuhkan untuk mereduksi gangguan dengan adanya perubahan *set point* maka dibutuhkan *filter* seperti berikut ini.

$$f_{(s)} = \frac{n\lambda s + 1}{(\lambda_s + 1)^n} \quad (2.29)$$

Filter tuning parameter akan membuat variasi kecepatan dari respon *closed loop system*, jika λ kecil maka respon dari sistem *closed loop* cepat, namun jika λ besar maka respon *closed loop* lebih *robust* (tidak terlalu *sensitive* dengan *model error*)

$$y(s) = g_p(s) \cdot q(s) \cdot r(s) = g_p(s) \cdot \tilde{g}_{p-(s)}^{-1} f(s) \cdot r(s) \quad (2.30)$$

Dimana;

$g_p(s)$ = *Gain process*

$q(s)$ = *Internal model control*

$r(s)$ = *Set point*

$\tilde{g}_p(s)$ = *Gain processmodel*

$y(s)$ = *Process output*

$f(s)$ = *Filter*

Jika *model* sempurna maka dapat diwakili oleh persamaan $g_p(s) = \tilde{g}_{p(s)}$, sehingga:

$$y(s) = \tilde{g}_{p+(s)} \cdot \tilde{g}_{p-(s)} \tilde{g}_{p-(s)}^{-1} f(s) \cdot r(s) \quad (2.31)$$

$$y(s) = \tilde{g}_{p+(s)} \cdot f(s) \cdot r(s) \quad (2.32)$$

Dimana;

$q(s)$ = *Internal model control*

$r(s)$ = *Set point*

$\tilde{g}_p(s)$ = *Gain processmodel*

$y(s)$ = *Process output*

$f(s)$ = *Filter*

Sistem *control level* pada dasarnya adalah suatu sistem pengendalian dengan orde satu. Begitu pula dengan *steam drum*, berdasarkan persamaan 2.29 dan 2.30 dapat disimpulkan bahwa, sistem pengendalian *level* pada *steam drum* adalah *system* orde satu. Sehingga berdasarkan *table Internal Model Control (IMC)* berbasis PID, maka dengan *gain proses first Order* maka akan didapatkan persamaan seperti dibawah ini :

$$g_p(s) = \frac{K_p}{\tau_s + 1} \quad (2.33)$$

Dimana;

K_p = *Gain plant*

τ_s = *Time constant*

$g_p(s)$ = *Gain process*

Jika *orde* dari suatu proses telah diketahui, maka dengan menggunakan Tabel IMC-PID akan didapatkan nilai K_c , T_i dan T_d . Berdasarkan Tabel IMC-PID, gain K_c , T_i dan T_d adalah sebagai berikut ini.

$$\tau_i = \tau_s \quad (2.34)$$

$$K_c = \frac{\tau_s}{\lambda \cdot K_p} \quad (2.35)$$

$$\text{Gain } CL(s) = \frac{1}{\lambda_s + 1} \quad (2.36)$$

Dimana;

K_c = *Gain controller*

T_i = *Integral time*

T_d = *Derivative time*

K_p = *Gain plant*

λ = *Parameter tuning filter (konstanta)*

τ_s = *Time constant plant*

Atau jika menggunakan persamaan yang lain pada tabel IMC-PID akan didapatkan persamaan seperti berikut ini.

$$\tau_i = \frac{2\tau_s\lambda - \lambda^2}{\tau_s} \quad (2.37)$$

$$K_c = \frac{2\tau_s - \lambda}{\lambda \cdot K_p} \quad (2.38)$$

$$\text{Gain } CL_{(s)} = \frac{r_s + 1}{(\lambda_s + 1)^2} \quad (2.39)$$

2.4 Respon Transient

Respon *transien* suatu sistem pengendali secara praktek selalu menunjukkan *osilasi* teredam sebelum mencapai keadaan tunaknya. Dalam menggolongkan karakteristik tanggapan *transient* suatu sistem pengendali terhadap masukan tangga satuan yaitu waktu tunda/*time delay* (td), waktu naik/*rise time* (tr), waktu puncak/*peak time* (tp), maksimum *overshoot* (Mp), *settling time* (ts) dimana^[5]:

1. Waktu tunda/*time delay* (td) adalah waktu yang digunakan oleh tanggapan untuk mencapai setengah nilai akhir untuk waktu yang pertama.
2. Waktu naik/*rise time* (tr) adalah waktu yang diperlukan oleh tanggapan untuk naik dari 10% menjadi 90%, 5% menjadi 95%, atau 0% menjadi 100% dari nilai akhir yang biasa digunakan. Untuk sistem atas redaman waktu naik yang biasa digunakan 10% menjadi 90%.
3. Waktu puncak (tp) adalah waktu yang diperlukan tanggapan untuk mencapai puncak pertama *overshoot*.
4. *Maksimum overshoot* (Mp) adalah nilai puncak kurva tanggapan diukur dari satuan. Apabila nilai akhir keadaan tunak tanggapannya jauh dari satu, maka biasa digunakan persen maksimum *overshoot* dan didefinisikan sebagai berikut :

$$\% \text{maximum overshoot} = \frac{c(tp) - c(\infty)}{c_{\infty}} \times 100\% \quad (2.40)$$

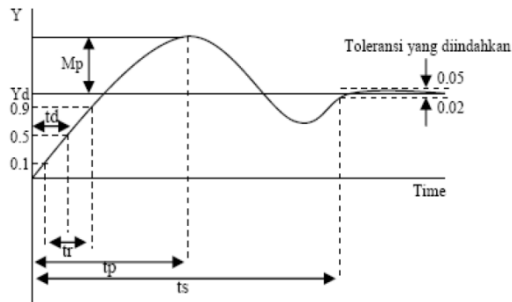
dimana : $c(tp)$: Keadaan pada waktu puncak

$c(\infty)$: Keadaan pada *setpoint*

Besarnya persen maksimum *overshoot* menunjukkan kestabilan relative dari system.

5. *Settling time* (ts) adalah waktu yang diperlukan untuk menanggapi kurva agar dapat mencapai dan tetap berada

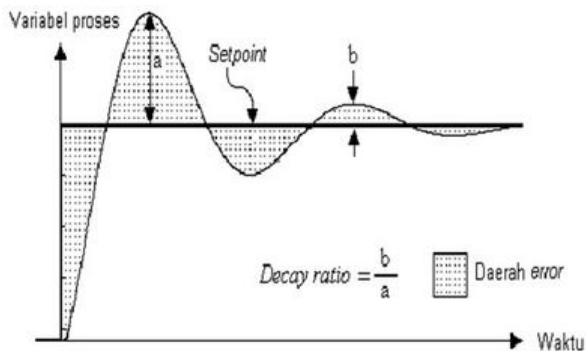
dalam gugus akhir ukuran yang disederhanakan dengan presentase mutlak harga akhirnya (biasanya 2% atau 5%).



Gambar 2.10 Grafik respontransient^[4]

6. *Integral absolute error* (IAE) merupakan penjumlahan nilai *error* dari respon *output* yang dihasilkan. Nilai IAE dapat ditentukan berdasarkan luas arsir kurva dari *respon* sistem yang dihasilkan. Sehingga dapat dikatakan bahwa penjumlahan nilai IAE diawali terhadap *respon* sistem ketika diberikan sebuah pengendali. Untuk menghitung nilai IAE dapat menggunakan persamaan dibawah ini,

$$IAE = \int_0^{\infty} e(t)dt \quad (2.41)$$

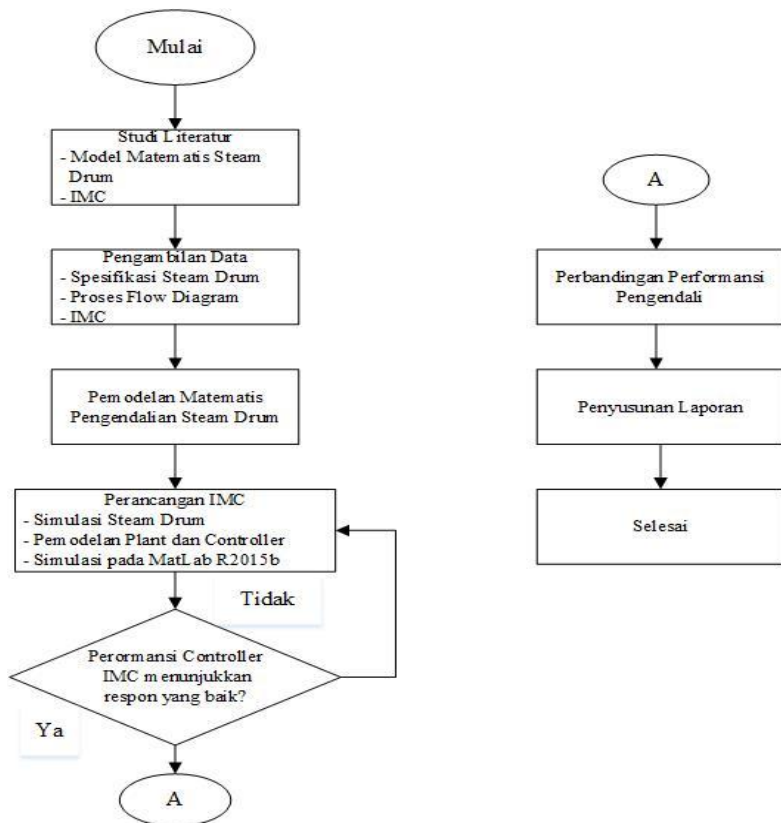


Gambar 2.11 Penentuan Nilai *Integral Absolute Error* (IAE) [Heriyanto, 2010]

BAB III METODOLOGI PENELITIAN

3.1 Alur Penelitian

Pada bab ini di jelaskan tentang tahapan pengerjaan tugas akhir mulai dari studi literatur, pemodelan sistem, perancangan sistem kendali *Internal Model Control* (IMC), simulasi dan uji performansi, serta penyusunan laporan tugas akhir. Beberapa tahapan tersebut dapat dijelaskan pada gambar di bawah ini.



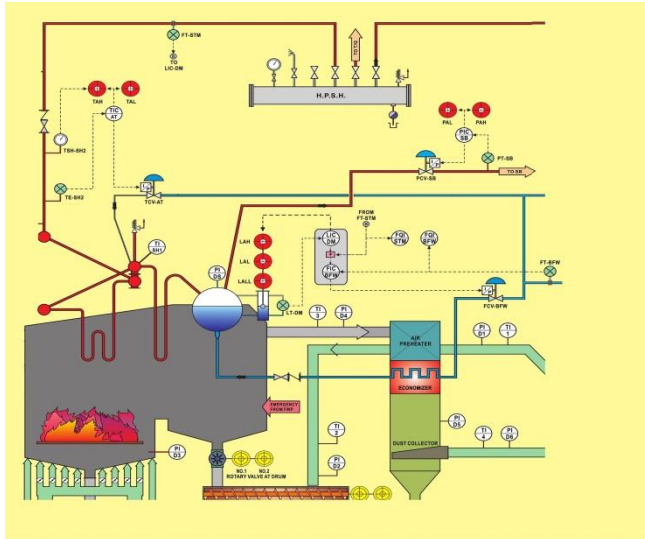
Gambar 3.1Flowchart Pengerjaan Tugas Akhir

3.2 Studi Literatur

Studi literatur dilakukan dengan mencari refrensi untuk *boiler drum* pada *plant*, proses dalam *boiler drum*, spesifikasi *boiler drum* dan kendali *Internal Model Control (IMC)*. Studi literature dilakukan dengan mencari jurnal-jurnal dan buku-buku yang berhubungan dengan *boiler drum* dan kendali IMC sebagai bahan pengerjaan tugas akhir.

3.3 Pengambilan Data

Pengambilan data *boiler drum* meliputi proses yang masuk dan kluar pada *boiler drum*. Data yang di peroleh nantinya akan digunakan sebagai acuan, untuk menyelesaikan perhitungan model matematis dari *palnt*. Secara garis besar, *boiler drum* berfungsi untuk menghasilkan *steam*, yang nantinya akan masuk ke *superheater* dan pada *superheater* uap yang di hasilkan harus benar-benar kering. Sehingga dapat di gunakan untuk memutar turbin.



Gambar 3.2 Process flow diagram steam drum

Table 3.1 Data *Input dan Output Steam drum*

No	Keterangan	in	out
1	Pressure	58 bar	47 bar
3	Temperature	230 °C	300 °C

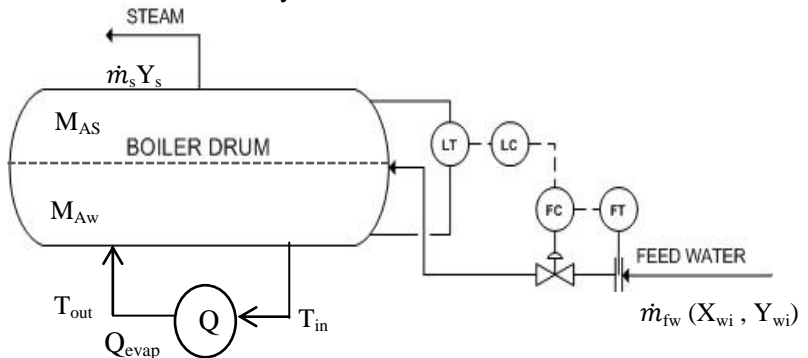
Data spesifikasi dimensi dari *boiler drum* diperoleh di tunjukkan pada table di 3.2.

Tabel 3.2 Data spesifikasi dimensi *steam drum*

No	Keterangan	Satuan (mm)
1	Diameter	1600
2	Panjang	8380

3.4 Model Matematis *Steam Drum*

Model matematis *boiler drum* mengacu pada proses pengendalian *level* pada *boiler drum*. *Boiler drum* memiliki masukan berupa H₂O *pure* dan *temperature*. Model matematis dilakukan dengan menggunakan hukum kesetimbangan *massa*, kompondan kesetimbangan *energy*. Dari proses *boiler drum*, di dapatkan *model* matematis yaitu,

**Gambar 3.3** P&ID *steam drum*

Kesetimbangan *massa* :

Yang terakumulasi = *input – output*

$$\frac{dm_A}{dt} = \dot{m}_{fw} - \dot{m}_s \quad (3.1)$$

$$\frac{d}{dt} (M_{as} + M_{aw}) = \dot{m}_{fw} - \dot{m}_s \quad (3.2)$$

$$\frac{d}{dt} M_{as} + \frac{d}{dt} M_{aw} = \dot{m}_{fw} - \dot{m}_s \quad (3.3)$$

Dimana;

$$m_A = (M_{as} + M_{aw}) \quad (3.4)$$

$$M_{as} = \rho_s A_s h_s \quad (3.5)$$

$$M_{aw} = \rho_w A_w h_w \quad (3.6)$$

Kesetimbangan komponen;

$$\frac{d}{dt} [M_{as} Y_s + M_{aw} X_{aw}] = \dot{m}_{fw} (X_{w \text{ in}} + Y_{s \text{ in}}) - \dot{m}_s Y_{s \text{ out}} \quad (3.7)$$

$$\frac{d}{dt} (M_{as} Y_s) + \frac{d}{dt} (M_{aw} X_{aw}) = \dot{m}_{fw} (X_{w \text{ in}} + Y_{s \text{ in}}) - \dot{m}_s Y_{s \text{ out}} \quad (3.8)$$

$$\frac{d}{dt} (M_{as} Y_s) = \dot{m}_{fw} (X_{w \text{ in}} + Y_{s \text{ in}}) - \dot{m}_s Y_{s \text{ out}} - \frac{d}{dt} (M_{aw} X_{aw}) \quad (3.9)$$

$$\frac{dM_{as}}{dt} = \frac{1}{Y_s} [\dot{m}_{fw} (X_{wi} + Y_{si}) - \dot{m}_s Y_{s \text{ out}} - X_{aw} \frac{dM_{aw}}{dt}] \quad (3.10)$$

Di kalikan dengan H (enthalpy), maka hasilnya;

$$H \frac{dM_{as}}{dt} = \frac{H}{Y_s} [\dot{m}_{fw} (X_{wi} + Y_{si}) - \dot{m}_s Y_{s \text{ out}} - X_{aw} \frac{dM_{aw}}{dt}] \quad (3.11)$$

Kesetimbangan Energi;

$$\frac{d}{dt}[M_{as}H + M_{aw}h] = \dot{m}_{fw}h_{fw} - \dot{m}_sH + Q_{evap} \quad (3.12)$$

$$H \frac{dM_{as}}{dt} + h \frac{dM_{aw}}{dt} = \dot{m}_{fw}h_{fw} - \dot{m}_sH + Q_{evap} \quad (3.13)$$

Substitusi persamaan (3.11) dan (3.13) maka;

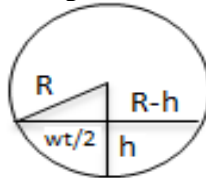
$$\frac{H}{Y_s} [\dot{m}_{fw}(X_{wi} + Y_{si}) - \dot{m}_sY_{sout} - X_{aw} \frac{dM_{aw}}{dt}] h \frac{dM_{aw}}{dt} = \dot{m}_{fw}h_{fw} - \dot{m}_sH + Q_{evap} \quad (3.14)$$

$$\left(h - \frac{HX_{aw}}{Y_s}\right) \frac{dM_{aw}}{dt} = \dot{m}_{fw} \left(h_{fw} - \frac{H(X_{wi} + Y_{si})}{Y_s}\right) - \dot{m}_s \left(H - \frac{HY_{sout}}{Y_s}\right) + Q_{evap} \quad (3.15)$$

Dimana $\frac{dM_{aw}}{dt}$ bisa tuliskan sebagai berikut,

$$\frac{dM_{aw}}{dt} = \rho_{aw} \frac{dV_{aw}}{dt} = \rho_w A_w \frac{dh_{lw}}{dt} \quad (3.16)$$

Dimana A_w bisa tuliskan sebagai,

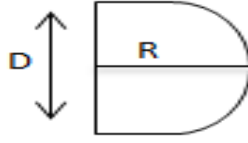


Gambar 3.4 luas *steam drum*

Dengan;

R = Sudut kemiringan

h = Ketinggian mula-mula *level air*



Gambar 3.5 luas *steam drums* samping

Dengan;

R = Jari-jari *steam drum* bagian samping

D = Batas tinggi lengkungan *steam drum* bagian samping

$$A_w = 2L \left[\frac{\pi H_{lw0} D}{2} \right] + \sqrt{R^2 - (R - (R - H_{lw0}))^2} \quad (3.17)$$

Sehingga,

$$\rho_w A_w \left(h - \frac{HX_{aw}}{Y_s} \right) \frac{dh_{lw}}{dt} = \dot{m}_{fw} \left(h_{fw} - \frac{H(X_{wi} + Y_{si})}{Y_s} \right) - \dot{m}_s \left(H - \frac{HY_{sout}}{Y_s} \right) + Q_{evap} \quad (3.18)$$

Dimana untuk \dot{m}_s bisa di tuliskan sebagai,

$$\dot{m}_s = \rho_s K \sqrt{H_{lw}} \quad (3.19)$$

(Dari katsuhiko ogata untuk aliran *steady state*)

Dimana;

Cd (coefficient discharge) = K, dengan nilai 0.6 - 0.85 m^{2.5}/s

Karena terdapat komponen *non-linier* maka dapat di linierisasi dengan deret Taylor, yaitu sebagai berikut;

$$K \sqrt{H_{lw}} = \left[\frac{dK \sqrt{H_{lw}}}{dh} \right]_{h=H_{lw0}} (H_{lw} - H_{lw0}) + \left[\frac{d^2 K \sqrt{H_{lw}}}{dh^2} \right]_{h=H_{lw0}} \frac{(H_{lw} - H_{lw0})^2}{2!} + \dots \quad (3.20)$$

$$K\sqrt{H_{lw}} = K\sqrt{H_{lw0}} + \frac{K}{2\sqrt{H_{lw0}}} (H_{lw} - H_{lw0}) - \frac{K}{8^3\sqrt{H_{lw0}}} (H_{lw} - H_{lw0})^2 + \dots \quad (3.21)$$

Hilangkan orde 2 dan orde tinggi, sehingga;

$$\rho_s K\sqrt{H_{lw}} = \rho_s K\sqrt{H_{lw0}} + \rho_s \frac{K}{2\sqrt{H_{lw0}}} (H_{lw} - H_{lw0}) \quad (3.22)$$

Kembali ke persamaan (3.18) dengan \dot{m}_s menggunakan persamaan (3.22)

$$\rho_w A_w \left(h - \frac{HX_{aw}}{Y_s} \right) \frac{dh_{lw}}{dt} = \dot{m}_{fw} \left(h_{fw} - \frac{H(X_{wi} + Y_{si})}{Y_s} \right) - \left(\rho_s K\sqrt{H_{lw0}} + \rho_s \frac{K}{2\sqrt{H_{lw0}}} (H_{lw} - H_{lw0}) \right) \left(H - \frac{HY_{s\ out}}{Y_s} \right) + Q_{evap} \quad (3.23)$$

$$\begin{aligned} & \rho_w A_w \left(h - \frac{HX_{aw}}{Y_s} \right) \frac{dh_{lw}}{dt} + \rho_s \frac{K}{2\sqrt{H_{lw0}}} \left(H - \frac{HY_{s\ out}}{Y_s} \right) H_{lw} \\ &= \dot{m}_{fw} \left(h_{fw} - \frac{H(X_{wi} + Y_{si})}{Y_s} \right) + Q_{evap} - \left(\rho_s K\sqrt{H_{lw0}} - \rho_s \frac{K}{2\sqrt{H_{lw0}}} H_{lw0} \right) \left(H - \frac{HY_{s\ out}}{Y_s} \right) \end{aligned} \quad (3.24)$$

Laplace persamaan (3.24);

$$\begin{aligned} & \rho_w A_w \left(h - \frac{HX_{aw}}{Y_s} \right) s H_{lw}(s) + \rho_s \frac{K}{2\sqrt{H_{lw0}}} \left(H - \frac{HY_{s\ out}}{Y_s} \right) H_{lw}(s) = \\ & \left(h_{fw} - \frac{H(X_{wi} + Y_{si})}{Y_s} \right) \dot{m}_{fw}(s) + Q_{evap}(s) - \left(\rho_s K\sqrt{H_{lw0}} - \rho_s \frac{K\sqrt{H_{lw0}}}{2} \right) \left(H - \frac{HY_{s\ out}}{Y_s} \right) \end{aligned} \quad (3.25)$$

$$\begin{aligned}
& \frac{\rho_w A_w \left(h - \frac{HX_{aw}}{Y_s} \right)}{\rho s \frac{K}{2\sqrt{H_{lw0}}} \left(H - \frac{HY_{s out}}{Y_s} \right)} \cdot s H_{lw}(s) + H_{lw}(s) \\
&= \frac{\left(h_{fw} - \frac{H(X_{wi} + Y_{si})}{Y_s} \right)}{\rho s \frac{K}{2\sqrt{H_{lw0}}} \left(H - \frac{HY_{s out}}{Y_s} \right)} \dot{m}_{fw}(s) + \\
& \quad \frac{Q_{evap}(s)}{\rho s \frac{K}{2\sqrt{H_{lw0}}} \left(H - \frac{HY_{s out}}{Y_s} \right)} \\
& \quad - \frac{\left(\rho s K \sqrt{H_{lw0}} - \rho s \frac{K \sqrt{H_{lw0}}}{2} \right) \left(H - \frac{HY_{s out}}{Y_s} \right)}{\rho s \frac{K}{2\sqrt{H_{lw0}}} \left(H - \frac{HY_{s out}}{Y_s} \right)}
\end{aligned} \tag{3.26}$$

Misal dari persamaan 3.26 di pisah, sehingga;

$$\tau = \frac{\rho_w A_w \left(h - \frac{HX_{aw}}{Y_s} \right)}{\rho s \frac{K}{2\sqrt{H_{lw0}}} \left(H - \frac{HY_{s out}}{Y_s} \right)} \tag{3.27}$$

$$k1 = \frac{\left(h_{fw} - \frac{H(X_{wi} + Y_{si})}{Y_s} \right)}{\rho s \frac{K}{2\sqrt{H_{lw0}}} \left(H - \frac{HY_{s out}}{Y_s} \right)} \tag{3.28}$$

$$k2 = \frac{1}{\rho s \frac{K}{2\sqrt{H_{lw0}}} \left(H - \frac{HY_{s out}}{Y_s} \right)} \tag{3.29}$$

$$k3 = \frac{\left(\rho s K \sqrt{H_{lw0}} - \rho s \frac{K \sqrt{H_{lw0}}}{2} \right) \left(H - \frac{HY_{s out}}{Y_s} \right)}{\rho s \frac{K}{2\sqrt{H_{lw0}}} \left(H - \frac{HY_{s out}}{Y_s} \right)} \tag{3.30}$$

Sehingga persamaan 3.26 menjadi,

$$\tau \cdot s H_{lw}(s) + H_{lw}(s) = k1 \dot{m}_{fw}(s) + k2 Q_{evap} - k3 \tag{3.31}$$

$$(\tau s + 1)H_{lw}(s) = k1 \dot{m}fw(s) + k2 Q_{evap} - k3 \quad (3.32)$$

$$H_{lw}(s) = \frac{k1}{\tau s + 1} \dot{m}fw(s) + \frac{k2}{\tau s + 1} Q_{evap}(s) - \frac{k3}{\tau s + 1} \quad (3.33)$$

$$\tau = \frac{\rho_w A_w \left(h - \frac{HX_{aw}}{Y_s} \right)}{\rho s \frac{K}{2\sqrt{H_{lw0}}} \left(H - \frac{HY_{s out}}{Y_s} \right)} \quad (3.34)$$

$$\tau = \frac{719,7 \times 47,08896 \left(1344 - 2749 \left(\frac{0,9566}{0,0434} \right) \right)}{58,65 \times \frac{0,85}{2\sqrt{0,8}} \left(2749 - 2749 \frac{1}{0,0434} \right)} \quad (3.35)$$

$$\tau = 1169 \text{ s} \quad (3.36)$$

$$k1 = \frac{\left(h_{fw} - \frac{H(X_{wi} + Y_{si})}{Y_s} \right)}{\rho s \frac{K}{2\sqrt{H_{lw0}}} \left(H - \frac{HY_{s out}}{Y_s} \right)} \quad (3.37)$$

$$k1 = \frac{\left(990 - 2803 \left(\frac{1}{0,0434} \right) \right)}{58,65 \times \frac{0,85}{2\sqrt{0,8}} \left(2749 - 2749 \frac{1}{0,0434} \right)} \quad (3.38)$$

$$k1 = 0,03600 \quad (3.39)$$

$$k2 = \frac{Q_{evap}}{\rho s \frac{K}{2\sqrt{H_{lw0}}} \left(H - \frac{HY_{s out}}{Y_s} \right)} \quad (3.40)$$

$$k2 = 0,03554 \quad (3.41)$$

$$k3 = \frac{\left(\rho s K \sqrt{H_{lw0}} - \rho s \frac{K \sqrt{H_{lw0}}}{2} \right) \left(H - \frac{HY_{s out}}{Y_s} \right)}{\rho s \frac{K}{2\sqrt{H_{lw0}}} \left(H - \frac{HY_{s out}}{Y_s} \right)} \quad (3.42)$$

$$k3 = \frac{(58,65 \times 0,85\sqrt{0,8} - 58,65 \frac{0,85\sqrt{0,8}}{2}) 58,65 \times \frac{0,85}{2\sqrt{0,8}} (2749 - 2749 \frac{1}{0,0434})}{58,65 \times \frac{0,85}{2\sqrt{0,8}} (2749 - 2749 \frac{1}{0,0434})} \quad (3.43)$$

$$k3 = 0,80028 \quad (3.44)$$

Sehingga persamaan model matematis *steam drum* menjadi,

$$H_{lw}(s) = \frac{k1}{\tau s + 1} \dot{m}fw(s) \quad (3.45)$$

$$H_{lw}(s) = \frac{0,03600}{1169s + 1} \dot{m}fw(s) \quad (3.46)$$

Sedangkan persamaan model matematis untuk *disturbance* menjadi,

$$gd = \frac{k2}{\tau s + 1} Q_{evap} - \frac{k3}{\tau s + 1} \quad (3.47)$$

$$gd = \frac{0,03554}{1169s + 1} - \frac{0,80028}{11690s + 1} \quad (3.48)$$

Dimana,

M_{as} = Massa akumulasi *steam*

M_{aw} = Massa akumulasi *water*

\dot{m} = Massa persatuan waktu (ton/jam)

M_s = Massa *steam*

M_w = Massa *water*

\dot{m}_{fw} = Massa *feed water* yang masuk

\dot{m}_s = Massa *steam* yang keluar

ρ_g = Massa *saturated gas* (kg/m³)

ρ_f = Massa *saturated liquid*

ρ_w = Masa jenis air

H_{lw} = *Level water* (m)

H_{lw0} = *Level air stady state* (ketinggian air mula-mula)

K = Cd = discharge coefficient of orifice plate m^{2.5}/s

A = Luas penampang *steam drum*

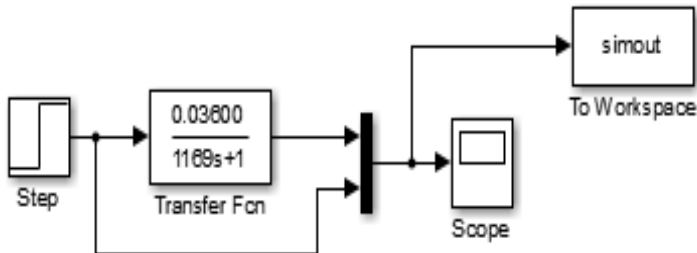
A_w = Luas penampang *steam drum water*

A_s = Luas penampang *steam drum steam*

h_{fw} = Enthalpy feed water(J)
 h = Enthalpy water
 H = Enthalpy steam (KJ/Kg)
 Q_{evap} = Kalor evaporator
 $X_{w \text{ in}}$ = Fraksifoodwater
 $Y_{s \text{ in}}$ = Fraksisteam input
 Y_s = Fraksi steam di dalam steam drum
 X_{aw} = Fraksi water di dalam steam drum
 τ = Time constant (s)

3.5 Uji Open Loop Gain Plant

Uji *open loop plant* dilakukan dengan menggunakan *software* MatLab R2015b. Uji *open loop* ini bertujuan untuk mengetahui respon dari *plant*, ketika belum menggunakan pengendali. Penyusunan *simulink* dilakukan seperti pada Gambar 3.6 berikut,

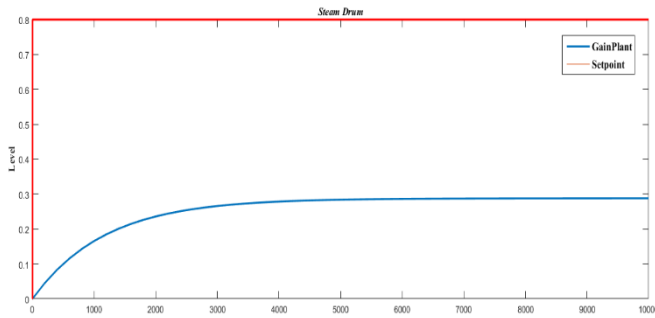


Gambar 3.6 Uji *open loop gain plant*

Pada sinyal *input* berisi nilai *set point* yaitu 0,8. Sedangkan pada *transferfunction* berisi pemodelan matematis *plant*. Pada *scope* untuk menunjukkan respon pada *plant*.

3.4.1 Hasil Simulasi Uji Open Loop Plant

Uji *open loop* bertujuan untuk mengetahui respon dari *plant* sebelum menggunakan *controller*. Simulasi dilakukan dengan menggunakan *software* MatLab R2015b. Hasil responpengendali *Internal Model Control* (IMC) di tunjukkan pada Gambar 3.7 sebagai berikut,

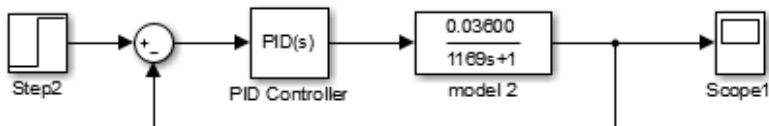


Gambar 3.7 Grafik Uji *open loop* Gain Plant

Dari Gambar 3.7 grafik uji *open loop plant* menunjukkan bahwa, respon pada *plant* akan terus bertambah sesuai dengan karakteristik dari *plant*. Pada grafik menunjukkan respon pada *plant* tidak dapat mencapai nilai *set point*, sehingga *plant* perlu diberi *control* agar mencapai *set point*.

3.6 Uji Close Loop Plant

Uji *open loop plant* di lakukan dengan menggunakan *software* Matlab R2015. Uji *close loop plant* ini bertujuan untuk mengetahui respon dari *plant*, ketika sudah menggunakan pengendali. Pada uji *close loop* ini pengendali yang di gunakan adalah PID. Penyusunan simulink dilakukan seperti Gambar 3.8 berikut,



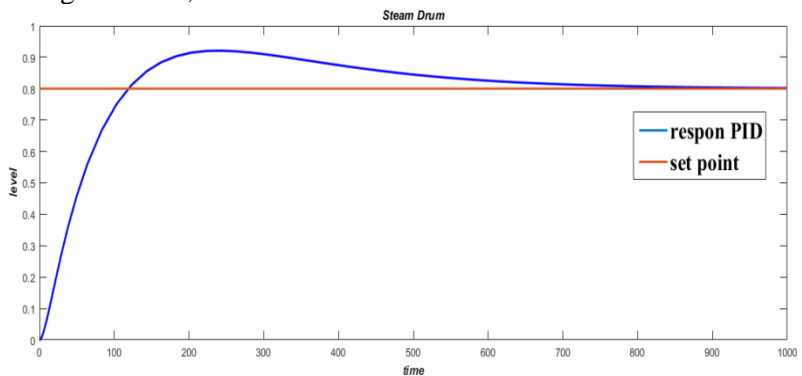
Gambar 3.8 Uji *close loop plant*

Pada sinyal *input* berisi niali *set point* yaitu 0,8. Sedangkan pada *transfer fuction* berisi pemodelan matematis *plant*, pada *disturbance* berisi pemodelan matematis selain *plant*. Pada *plot*

pengendali PI berdasarkan pada *real plant*, maka nilai $P = 28,0887$ dan $I = 2,005435$. Pada *scope* untuk menunjukkan respon pada *plant*.

3.6.1 Hasil Simulasi Uji Close Loop Plant

Uji *open loop plant* di lakukan dengan menggunakan *software* MatLab R2015b. Uji *close loop plant* ini bertujuan untuk mengetahui respon dari *plant*, ketika sudah menggunakan pengendali. Pada uji *close loop* ini pengendali yang di gunakan adalah PI. Penyusunan simulink dilakukan pada Gambar 3.9 sebagai berikut,



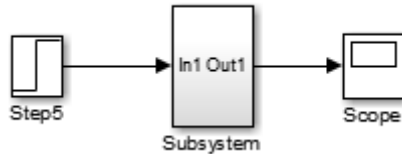
Gambar 3.9 Grafik Uji *close loop Plant*

Pada sinyal input berisi nilai *set point* yaitu 0.8. Sedangkan pada *transfer function* berisi pemodelan matematis *plant*, pada *disturbance* berisi pemodelan matematis selain *plant*. Pada plot pengendali PI berdasarkan pada *auto tuning*, maka nilai $P = 28,0887$ dan $I = 2,005435$. Pada *scope* untuk menunjukkan respon pada *plant*.

3.7 Uji Disturbance

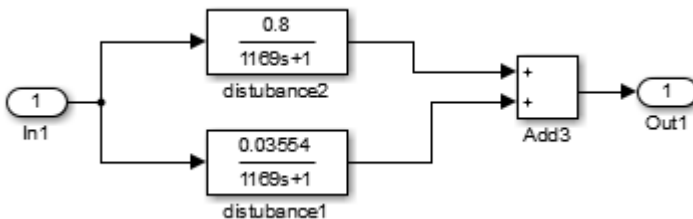
Uji *Disturbance* berfungsi untuk mengetahui pengaruh yang pada *plant*. Model matematis *disturbance* diketahui pada persamaan 3.48, yaitu meliputi *mass flow* dan Q_{evap} . Uji *disturbance* ini dilakukan, bertujuan untuk mengetahui seberapa

besar pengaruh *disturbance* terhadap *level steam drum*. Penyusunan simulink ditunjukkan pada Gambar 3.10 sebagai berikut,



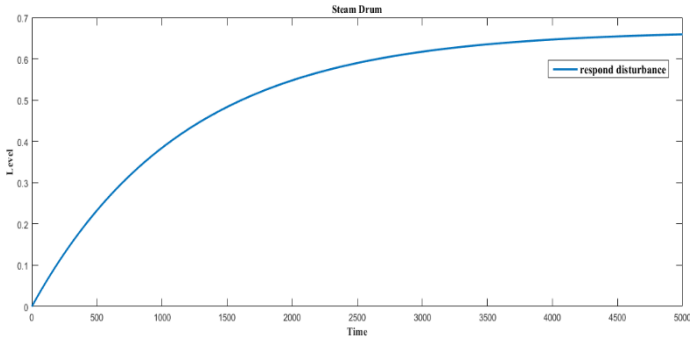
Gambar 3.10 Uji *Disturbance*

Pada subsystem, berisi nilai dari model matematis *disturbance* pada persamaan 3.48. *subsystem* ditunjukkan pada Gambar 3.11 sebagai berikut,



Gambar 3.11 Uji *Disturbance*

Pada *subsystemdisturbance* didapat dari persamaan 3.48, dimana pemanasan (Q_{evap}) dan *output steam drum* ($\dot{m}_s Y_s$). Pada *disturbance* ini yang nantinya akan mempengaruhi respons *system* pada pengendali. Hasil simulasi *responsdisturbance* ditunjukkan pada gambar 3.1 sebagai berikut,



Gambar 3.12 Hasil Simulasi Uji *Disturbance*

3.7 Pengendali IMC

3.7.1 Uji Pengendali Tanpa *Disturbance*

Perancangan *control Internal Model Control* (IMC) ini, dilakukan dengan merancang fungsi *transfer* pengendali dari fungsi *transfer plant*. Sedangkan fungsi *transfer plant* di dapatkan dari pemodelan matematis *plant*. Pada pemodelan matematis di dapat juga *disturbance*. Terdapat 2 *disturbance* yaitu, dari proses pemanasan (Q_{evap}) dan *output steam drum* ($\dot{m}_s Y_s$). Persamaan pengendali IMC adalah,

$$q(s) = \tilde{g}_{-p}^{-1} f(s) = \tilde{g}_{-p}^{-1}(s) \frac{1}{\lambda s + 1} \quad (3.49)$$

Dari fungsi *transfer plant* didapatkan nilai pengendali IMC sebesar,

$$\tilde{g}_{-p}^{-1} = \frac{1169s + 1}{0.03600} \quad (3.50)$$

Sehingga persamaan pengendali IMC menjadi,

$$q(s) = \frac{1169s + 1}{0.03600} \frac{1}{\lambda s + 1} \quad (3.51)$$

Parameter *tunning* IMC yaitu λ , berfungsi untuk mendapatkan respon yang terbaik. Respon tersebut yang nantinya akan di bandingkan dengan pengendali PID. Nilai λ yang di gunakan sebanyak 4 variasi berdasarkan persmaan 2.12, 2.13, 2.14 dan 2.15. Maka fungsi transfer pengendali IMC adalah sebagai berikut,

$$\lambda = 0,1 \text{ (Rivera)} \quad (3.52)$$

$$q(s) = \frac{1169s + 1}{0,03600} \frac{1}{0,1s + 1} \quad (3.53)$$

$$q(s) = \frac{1169s + 1}{0,0036s + 0,03600} \quad (3.54)$$

$$\lambda = 0,8 \text{ (Rivera)} \quad (3.55)$$

$$q(s) = \frac{1169s + 1}{0,03600} \frac{1}{0,8s + 1} \quad (3.56)$$

$$q(s) = \frac{1169s + 1}{0,0288s + 0,03600} \quad (3.57)$$

$$\lambda = 1 \text{ (Sigurd Skogestad)} \quad (3.58)$$

$$q(s) = \frac{1169s + 1}{0,03600} \frac{1}{1s + 1} \quad (3.59)$$

$$q(s) = \frac{1169s + 1}{0,03600s + 0,03600} \quad (3.60)$$

$$\lambda = 1,5 \text{ (Wyne Bequette)} \quad (3.61)$$

$$q(s) = \frac{1169s + 1}{0,03600} \frac{1}{1,5s + 1} \quad (3.62)$$

$$q(s) = \frac{1169s + 1}{0,064s + 0,03600} \quad (3.63)$$

$$\lambda = 0.001 \quad (3.64)$$

$$q(s) = \frac{1169s + 1}{0,03600} \frac{1}{0,05s + 1} \quad (3.65)$$

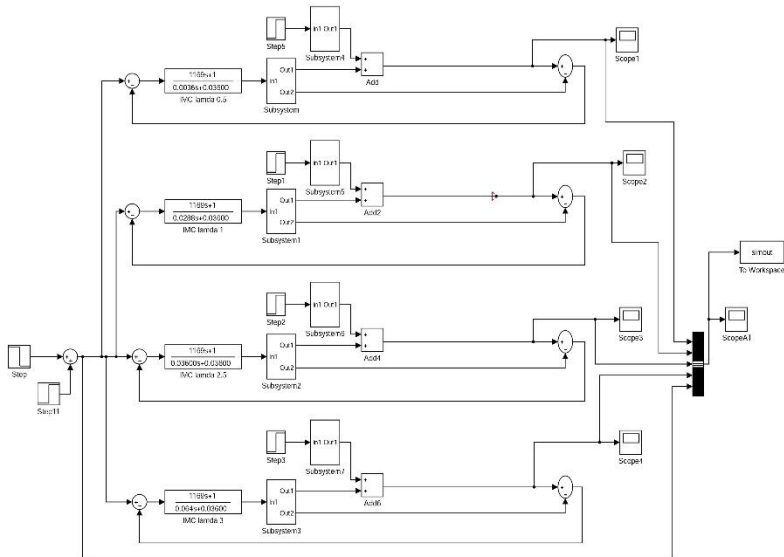
$$q(s) = \frac{1169s + 1}{0,0018s + 0,03600} \quad (3.66)$$

$$\lambda = 0,001 \quad (3.67)$$

$$q(s) = \frac{1169s + 1}{0,03600} \frac{1}{0,05s + 1} \quad (3.68)$$

$$q(s) = \frac{1169s + 1}{0,0018s + 0,03600} \quad (3.69)$$

Simulasi menggunakan *software* MatLab R2015b untuk mendapatkan grafik respon. Susunan pada *Simulink* didasarkan dengan diagram pengendali IMC. Penyusunan IMC tanpa adanya *disturbance* pada MatLab R2015b, ditunjukkan seperti pada Gambar 3.10 berikut,



Gambar 3.13 Perancangan Uji Pengendali Tanpa *Disturbance*

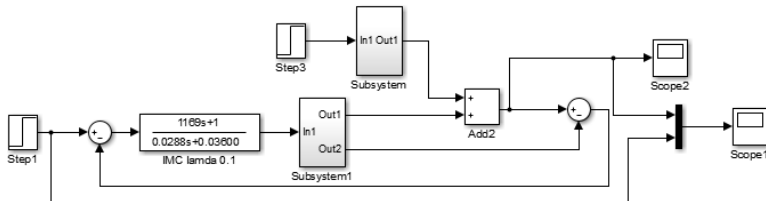
Pada uji pengendali pada *software* MatLab R2015b, *subsystem* berisi nilai dari fungsi *transfer* yang di dapat dari pemodelan matematis *plant*.

Perbandingan respon *system* dilakukan dengan membandingkan respon dari *control* IMC. Parameter respon yang digunakan untuk melihat performansi pengendali adalah *maximum overshoot*, *settling time*, *integral absolute error* (IAE) dan *error steady state* (Ess). Respon terbaik nantinya yang akan di pakai, dan dibandingkan dengan pengendali dengan menggunakan metode yang lain.

3.8.2 Uji Pengendali Dengan *Disturbance*

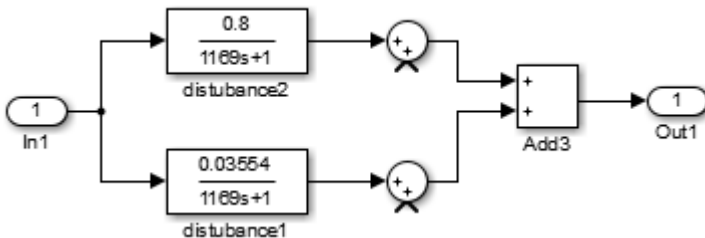
Uji pengendali dengan *disturbance* ini bertujuan, untuk mengetahui respon pengendali apabila mendapat *disturbance* dari proses. *Disturbance* yang dimasukkan adalah pemanasan (Q_{evap}) dan *output steam drum* ($\dot{m}_s Y_s$). Penyusunan pada *software*

MatLab R2015b dilakukan seperti pada Gambar 3.14 sebagai berikut,



Gambar 3.14 Simulasi Uji Pengendali Dengan *Disturbance*

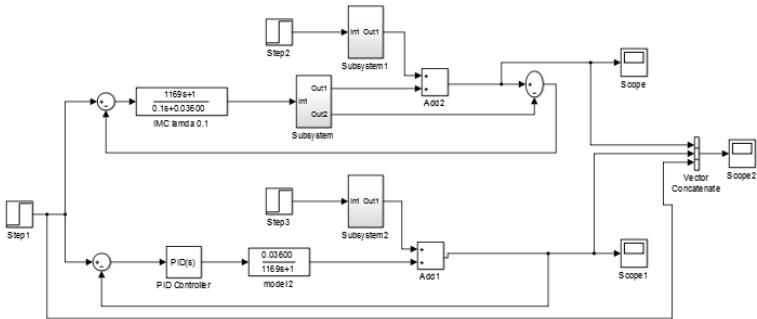
Pada *subsystemdisturbance*, berisi nilai dari fungsi transfer *disturbance* dari model matematis pada persamaan 3.48 dilakukan seperti pada gambar 3.14 sebagai berikut.



Gambar 3.15 Simulasi *Subsystem Disturbance*

3.9 Perbandingan Respon Sistem

Perbandingan respon *system* dilakukan dengan membandingkan respon dari pengendali IMC, dengan pengendali PI yang menggunakan *disturbance*. Parameter *tuningcontroller* PI pada *real plant* yang di gunakan memiliki nilai $P = 28,0887$ dan $I = 2,0054$ untuk mendapatkan respon *system*, simulasi dilakukan dengan menggunakan *software* MatLab R2015b. Pengendali dari IMC yang di gunakan adalah pengendali dengan nilai λ terkecil. Rancangan pengendali pada *simulink* di tunjukan pada Gambar 3.11 berikut,



Gambar 3.16 Simulasi Perbandingan Pengendali IMC dan PI

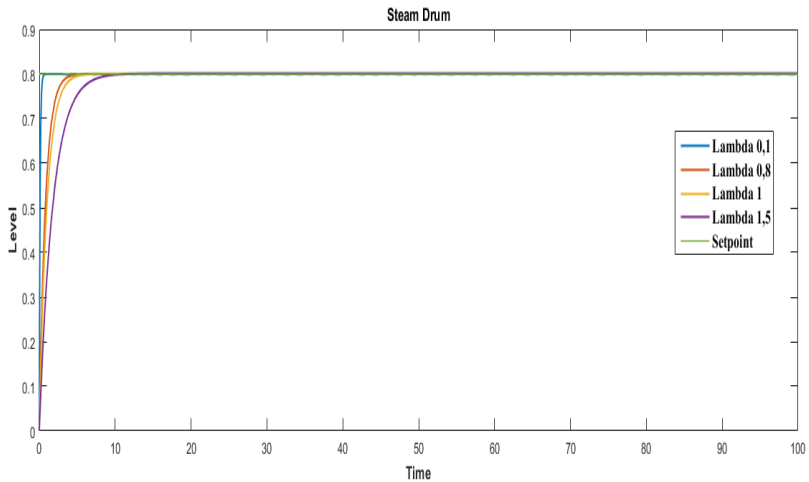
Pada simulasi perbandingan pengendali IMC dengan PI, pengendali IMC menggunakan respon yang paling baik, dengan nilai *tuning* λ 0,1. Perbandingan respon ini juga bertujuan untuk mengetahui pengendali mana yang terbaik, dan juga cocok dipasang di dunia industri.

BAB IV ANALISA DATA DAN PEMBAHASAN

4.1 Hasil Perancangan Pengendali IMC

4.1.1 Hasil Perancangan Pengendali Tanpa *Disturbance*

Simulasi pengendali IMC ini bertujuan untuk mendapatkan respon *system*, dengan parameter *tuning* pengendali terbaik. Simulasi dilakukan dengan menggunakan *software* MatLab R2015b. Hasil respon pengendali IMC tanpadisturbance dengan 4 *lambda* yang berbeda, di tunjukkan pada gambar 4.3 sebagai berikut,



Gambar 4.1 Respon Pengendali IMC Tanpa Disturbance

Parameter respon yang di gunakan adalah *settling time*, *maximum overshoot*, *Integral Absolute Error* (IAE) dan *Error steady State*. Parameter yang di peroleh di tunjukkan pada Tabel 4.1 sebagai berikut,

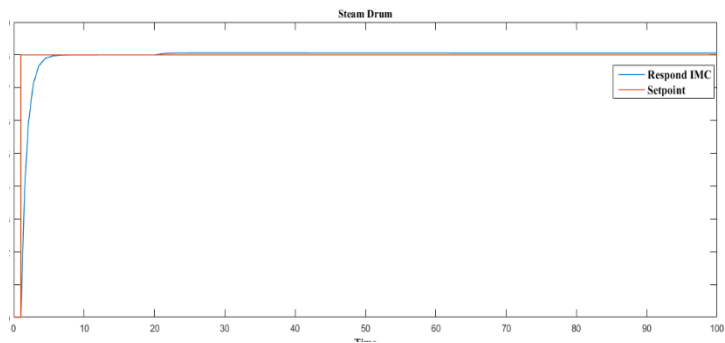
Tabel 4.1Parameter Pengendali IMC Tanpa *Disturbance*

Parameter <i>Tuning</i> IMC	<i>Maximum</i> <i>Overshoot</i> (%)	<i>Settling</i> <i>Time</i> (s)	IAE	Ess (%)
$\lambda = 0,1$	0	7,545023	8,764348	0,00075
$\lambda = 0,8$	0	55,90	14,78918	0,0725
$\lambda = 1$	0	82,05037	17,15872	0,062
$\lambda = 1,5$	0	192,5534	24,93502	0,71

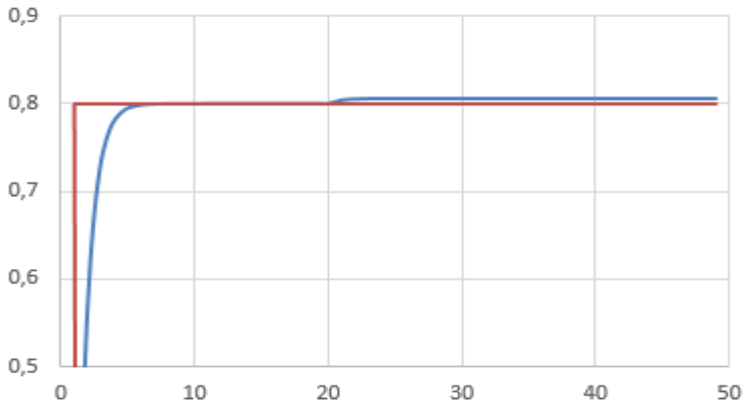
Dari hasil parameter responpengendali IMC dapat diketahui bahwa, respon *system* dengan menggunakan pengendali IMC dapat mencapai *set point*. Parameter *tuning* IMC yang terbaik adalah pengendali IMC dengan nilai *lambda* terkecil, yaitu $\lambda = 0,1$. Nilai *Maximum overshoot* didapatkan dari selisih antara puncak osilasi dan *set point*, kemudian di bagi dengan nilai *set point* dan di kalikan 100%. Pada uji pengendali tanpa adanya *disturbance* ini tidak ada *maximum overshoot*.

4.1.2 Hasil Perancangan Pengendali Dengan *Disturbance*

Pemasangan *disturbance* dilakukan degan menggunakan nilai paraeter *tuning* λ sebesar 0,1. karena nilai $\lambda = 0,1$ memiliki respon paling baik. Hasil respon pengendali dengan menggunakan *disturbance* ditunjukan pada Gambar 4.2.



Gambar 4.2Respon Pengendali IMC Dengan *Disturbance*



Gambar 4.3 Respon Pengendali IMC Dengan *Disturbance* (Perbesar 3x)

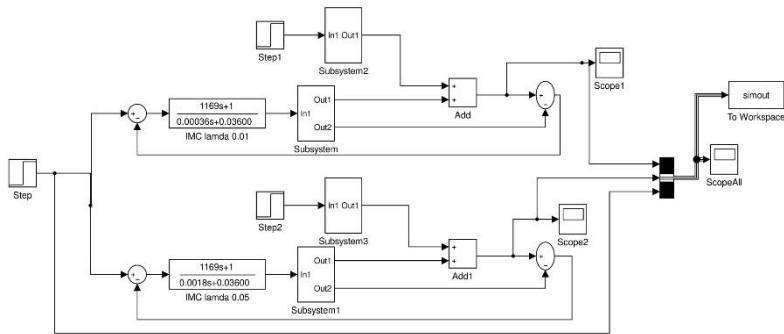
Dengan penambahan *disturbance* maka pada waktu ke-20 respon sistem akan terjadi gangguan, sehingga menimbulkan *maximum overshoot* sebesar 0,00625. hal ini dikarenakan adanya gangguan yang terjadi dari pemanasan (Q_{evap}) dan *output steam drum* ($\dot{m}_s Y_s$).

Tabel 4.2 Parameter Pengendali IMC Dengan *Disturbance*

Parameter Tuning IMC	Maximum Overshoot (%)	Settling Time (s)	IAE	Ess (%)
$\lambda = 0,1$	0,00625	7,545023	8,764348	0,000025

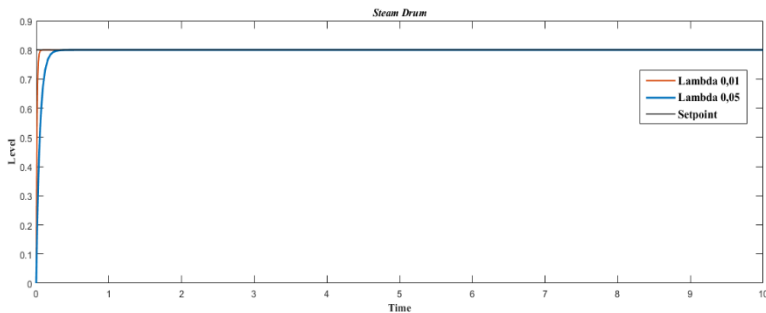
4.2 Hasil Perancangan IMC Dengan *Tuning* Acak

Perancangan ini dilakukan bertujuan untuk mengetahui parameter *tuning lambda* apabila dipilih secara acak. Artinya pada *tuning lambda* tidak mengikuti parameter *tuning* seperti pada persamaan 3.52, 3.55, 3.58, 3.61. parameter yang dipilih yaitu 0,01 dan 0,05. simulasi dilakukan dengan menggunakan MatLab R2015b seperti pada Gambar 4.3 sebagai berikut,



Gambar 4.4 Perancangan Pengendali IMC *Tuning* $\lambda = 0,01$ dan $\lambda = 0,05$

Pada uji pengendali IMC tuning acak menggunakan *software* MatLab R2015b, *subsystem* berisi nilai dari fungsi *transfer* yang di dapat dari pemodelan matematis *plant*. Hasil perancangan simulasi pengendali ditunjukkan pada Gambar 4.4 sebagai berikut,



Gambar 4.5 Hasil Perancangan pengendali IMC *Tuning* $\lambda = 0,01$ dan $\lambda = 0,05$

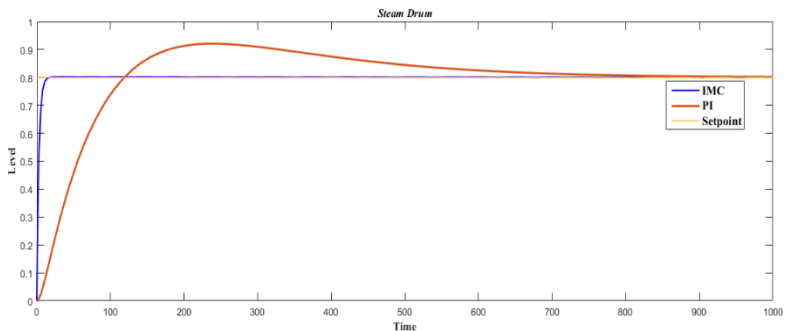
Parameter respon yang di gunakan adalah *settling time*, *maximum overshoot*, *Integral Absolute Error* (IAE) dan *Error steadyState*. Parameter yang di peroleh di tunjukkan pada Tabel 4.1 sebagai berikut,

Tabel 4.3Parameter *Tuning* IMC $\lambda = 0,01$ dan $\lambda = 0,05$

Parameter <i>Tuning</i> IMC	Maximum Overshoot (%)	Settling Time (s)	IAE	Ess (%)
$\lambda = 0,01$	0,00025	4,265023	5,914346	0,000015
$\lambda = 0,05$	0,000193	2,731053	3,529142	0,000009

4.3 Hasil Perbandingan Respon Pengendali

Perbandingan *responsystem* dilakukan dengan membandingkan respon dari pengendali IMC, dengan pengendali PI yang menggunakan *disturbance*. Parameter *tuning* pengendali PI pada *real plant*, yaitu memiliki nilai $P = 28,0887$ dan $I = 2,0054$ untuk mendapatkan respon *system*. Sedangkan simulasi dilakukan dengan menggunakan *software* MatLab R2015b. pengendali dari IMC yang di gunakan adalah pengendali dengan nilai λ terkecil (0,1). Rancangan pengendali pada *simulink* di tunjukan pada Gambar 4.6 berikut,

**Gambar 4.6** Respon Perbandingan Pengendali IMC dan PID

Parameter respon yang di gunakan adalah *settlingtime*, *maximum overshoot*, *Integral Absolute Error* (IAE) dan *Error steady State* (Ess). Parameter yang di peroleh di tunjukan pada Tabel 4.2 sebagai berikut,

Tabel 4.4Perbandingan Parameter Respon Pengendali

<i>Controller</i>	<i>Maximum Overshoot(%)</i>	<i>Settling Time (s)</i>	IAE	Ess (%)
IMC	0,00625	7,545023	8,764348	0,000025
PI	7,131	31,5412	17,8329	0,01785

Pada hasil tabel 4.2 menunjukkan bahwa, perbandingan antara respon pengendali IMC memiliki nilai *maximum overshoot*, *settling time*, IAE, Ess lebih kecil dari pada pengendali PID.

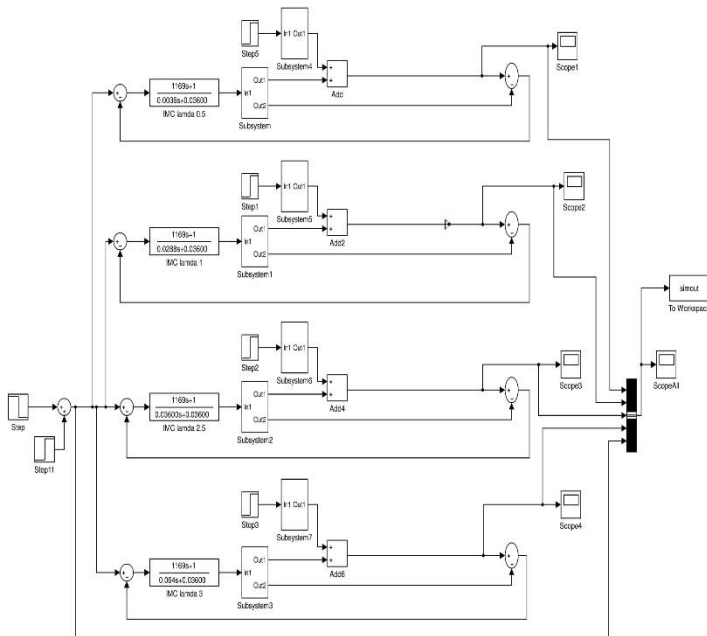
Tabel 4.5Perbandingan Parameter *Tuning* Pengendali

<i>Controller</i>	IMC (λ)	Kp (s)	Ti
IMC	0,1	-	-
PI	-	28.,887	2,0054

Pada tabel 4.3 menunjukkan *tuning* pengendali yang di gunakan pada simulai MatLab R2015b. Dimana untuk IMC hanya ada satu *tuning*, yaitu menggunakan *lambda* sebesar 0,1. Sedangkkn pada pengendali PI menggunakan 2 parameter *tuning* yaitu Kp dan Ti.

4.4 Pengujian *Tracking Set Point* Pengendali IMC

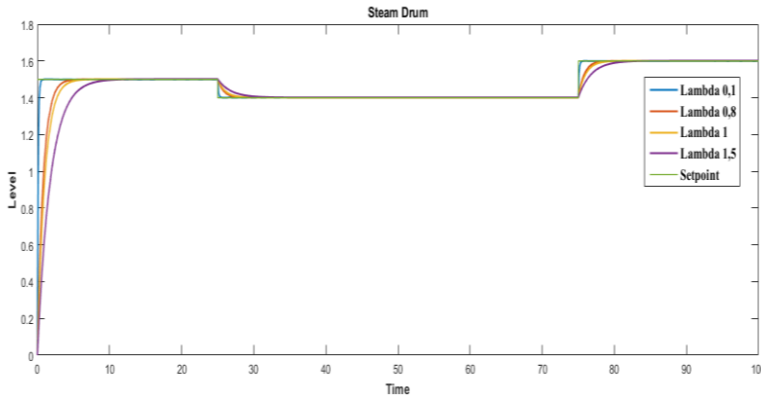
Uji *tracking set point* bertujuan untuk mengetahui performansi pengendali IMC, apabila *set point* pada proses dirubah *set point real plant* adalah 0,8 m. Untuk mendapatkan *respon* pengendali, Simulasi *trackingset point* dilakukan dengan *software* MatLab R2015b. Rancangan *tracking set point* pada *simulink* di tunjukkan pada Gambar 4.7 sebagai berikut,



Gambar 4.7 Simulasi Uji *TrackingSet Point* Pengendali IMC Dengan Variasi 4 *Lambda*

Pada simulasi uji *tracking set point*, pada dasarnya sama seperti pada simulasi uji *close loop plant*. Namun perbedaannya ada pada *set point* yang di berikan, sehingga *input* di isi lebih besar atau lebih kecil dari parameter *set point*.

Hasil respon uji *tracking set point* pengendali IMC di tunjukkan pada Gambar 4.8 sebagai berikut,



Gambar 4.8 Hasil Respon Uji *Tracking Set Point* Pengendali IMC Dengan Variasi 4 Λ

Dari hasil respon uji *tracking set point*, didapatkan bahwa pengendali IMC mampu mencapai *set point* ketika *set point* dirubah. Ketika pada *set point* di beri nilai sebesar 1,5, maka pengendali mampu mencapai *set point*. Kemudian ketika *set point* di turunkan nilainya sebesar 1,4, maka pengendali mampu mencapai *set point*.

4.5 Pembahasan

Perancangan pengendalian *level steam drum* pada penelitian ini adalah dengan menggunakan metode *Internal Model control*. Respon dari pengendalian dengan metode IMC dilakukan dengan menentukan nilai parameter *tuning* IMC, yaitu λ . Simulasi dari *model* sistem *steam drum* menggunakan pemodelan matematis *plant*. Pada pemodelan *plant*, *model plant* menghasilkan pemodelan matematis *plant* dan *disturbance*. Dari respon *control* IMC, menunjukkan bahwa pengendali dengan parameter nilai $\lambda = 0,1$ mempunyai respon yang paling baik. Dengan nilai *maximumovershoot* paling kecil yaitu 0,00625%. *Maximumovershoot* yang ditimbulkan cari pengendali IMC, yaitu dengan adanya *disturbance* dari pemodelan matematis *plant*. Selain itu, Q_{evap} akan mempengaruhi pada *level steam drum*,

begitu juga dengan $\dot{m}_s Y_s$. jika semakin besar nilai *disturbance*, maka *level* pada *steam drum* akan cepat turun. Dasar teori dari parameter *tuning* IMC menyatakan bahwa, untuk mencari respon yang paling cepat maka, nilai λ harus diperkecil. Respon dari pengendali IMC dengan nilai λ paling kecil, mempunyai nilai *settling time* kecil, yaitu 7,545023s. Mempunyai nilai *Error Steady State* dan *Integral Absolute Error* yang kecil juga, yaitu $Ess = 0,000025$ dan $IAE = 8,764348$. Nilai $\lambda = 0,1$ digunakan untuk perbandingan performansi pengendali yang telah digunakan, yaitu PID. Hasil uji performansi pengendali menunjukkan bahwa, respon pengendali IMC memiliki respon yang lebih baik, dari pada pengendalian PID dengan nilai *maximum overshoot* sebesar 15,131%. Nilai *settling time* yang di tunjukkan pada pengendali PI sebesar 31,5412s. Kemudian untuk nilai $Ess = 0,01785$ dan $IAE = 17,8329$. Parameter pengendalian pada IMC memiliki parameter yang baik dibandingkan dengan PI.

(Halaman ini sengaja dikosongkan)

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Dari penelitian tugas akhir ini dapat disimpulkan bahwa:

1. Perancangan pengendalian *level* pada *steam drum* di PTPN X krebung dengan metode *Internal Model Control*(IMC), dapat dilakukan dengan memperkecil nilai parameter *tuning* IMC (*lambda*) untuk mendapatkan respon terbaik.
2. Pengendalian IMC dengan nilai parameter *tuning* IMC (*lambda*) sebesar 0,1 memiliki performansi nilai *maximumover shoot* sebesar 0,00625, *settling time* sebesar 7,545023s, *Error Steady State* sebesar 0.000025dan*Integral Absolute error* sebesar 8,764348.

5.2 Saran

Dari hasil penelitian tugas akhir ini terdapat beberapa saran untuk pengembangan penelitian selanjutnya antara lain:

1. Dapat menambahkan komposisi pada *output steam drum*, karena pada penelitian ini komposisi pada *output steam drum* dianggap uap kering dan siap pakai.
2. Dapat dilakukan perancangan pengendalian IMC sampai mengetahui *tunning* IMC berbasis PID, karena pada penelitian kali ini hanya memakai pengendali IMC.

(Halaman ini sengaja dikosongkan)

DAFTAR PUSTAKA

- [1] K. Ogata and Y. Yang, *“Modern Control Engineering”*, Prentice-Hall Englewood Cliffs, NJ, 1970.
- [2] B. W. Bequette, *“Process control: modeling, design, and simulation”*, Prentice Hall Professional, 2003.
- [3] E. Marlin. Thomas, *“Design Process And Control System for Dinamic Performance”*, McGraw-Hill Higher Education.
- [4] Rivera. D. Daniel, Morari. M, Skogetad. S *“Internal Model Control. 4. PID Controller Design”* California Institute of Technology, Pasadena, California 91125.
- [5] Hidayat. A. Moammar, *“Perancangan Sistem Pengendalian Level Menggunakan Metode Internal Model Control (IMC) Pada Daerator 101U Plant Amoniak PT. Petrokimia Gresik”*, 2016.
- [6] Saputra. R. Fandi, *“Perancangan Sistem Pengendalian Level Liquid Oxygen Pada Rectifying Column Di PT. Samator Gas Industri Menggunakan Metode Internal Model Control”*, Teknik Fisika, FTI ITS.
- [7] A. Kusnadi, I. Setiawan, *“Perancangan Pengontrol Dan Analisis Respon Pada Sistem Internal Model Control (IMC),”* 2011.
- [8] S. Skogestad, *“Simple Analytic Rules For Model Reduction And PID Controller Tuning”*, Elsevier, 2003.
- [9] H. Noor, *“ Comparison On First-Order Filter And Higher-Order Filter In An Internal Model Control (IMC) Design*, University Teknologi Malaysia, 2016.
- [10] I. G. Horn, J. R. Arulandu, C. J. Gombas, J. G. VanAntwerp, and R. D. Braatz, "Improved filter

design in internal model control," *Industrial & engineering chemistry research*, vol. 35, pp. 3437-3441, 1996.

- [11] I. Setiawan, "Kontrol PID untuk Proses Industri," 2008.
- [12] M. T. Tham, "*Internal model control*," Introduction To Robust Control-[en ligne]-Disponible sur<
<http://lorien.ncl.ac.uk/ming/robust/imc.pdf>>(Consulté le 12.11. 2010), 2002.

BIODATA PENULIS



Penulis dilahirkan di Sidorjo, pada tanggal 05 Maret 1993. Penulis menempuh pendidikan formal di MI Miftahul Ulum Mulyodadi, MTsN Krian, SMA Wachid Hasyim 2 Taman, D3 Metrolog & iInstrumentasi ITS. Kemudian penulis melanjutkan studi diProgram Studi S-1Teknik Fisika FTI – ITS Surabaya pada tahun 2014.

Dengan ketekunan dan semangat yang tinggi untuk terus belajar dan berusaha, penulis akhirnya berhasil menyelesaikan pengerjaan tugas akhir. Semoga dengan pengerjaan tugas akhir ini mampu memberikan kontribusi positif bagi dunia pendidikan serta pengembangan untuk penelitian selanjutnya. Berikut akan dilampirkan alamat email dan juga nomor tlp dari penulis, bagi pembaca yang akan bertanya bisa langsung menghubungi contact yang ada di bawah.

Email : gebyhamzah@gmail.com
No. Hp : 081212414358